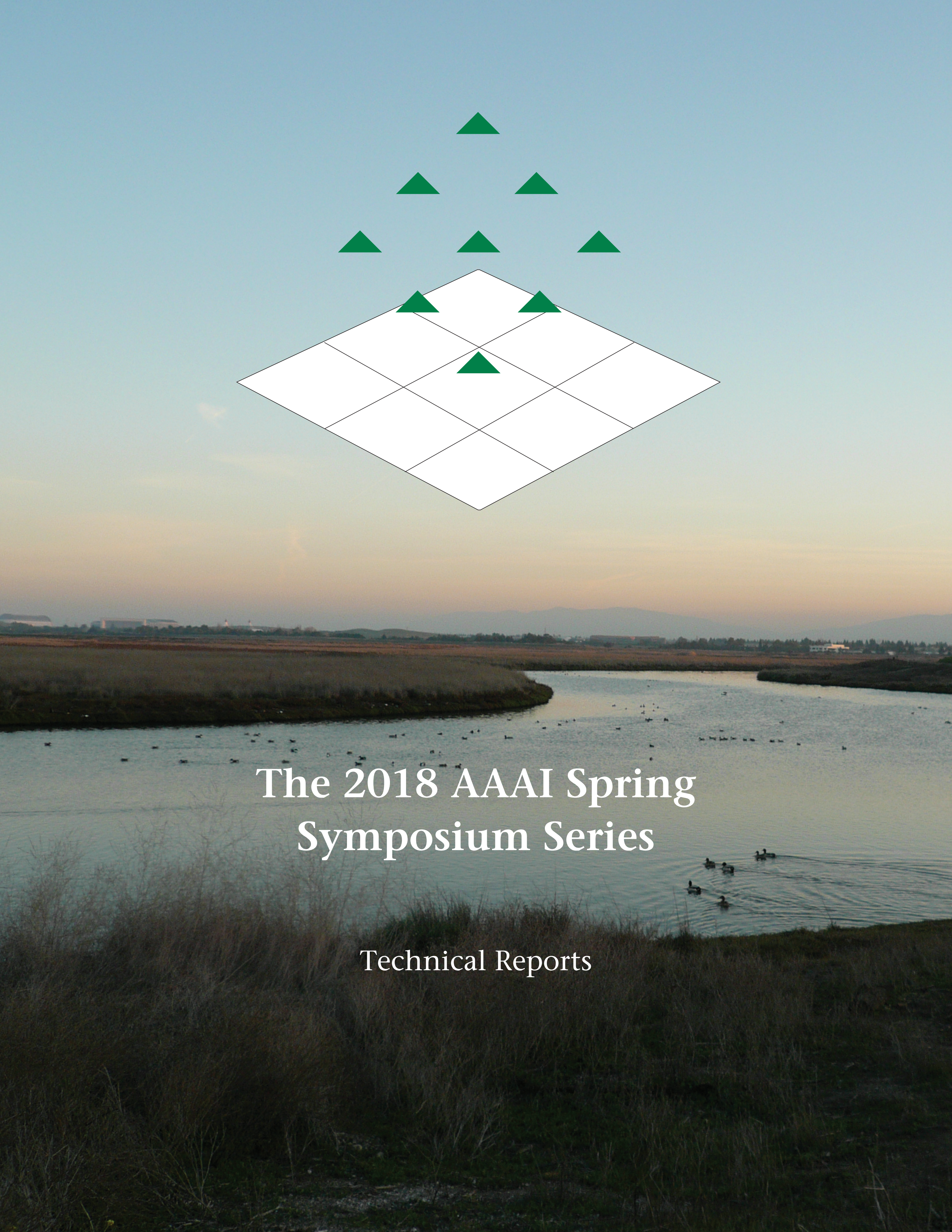


The 2018 AAAI Spring Symposium Series

Technical Reports

The 2018 AAAI Spring Symposium Series

Technical Report
SS-18

(Collected in One Volume)

AAAI Press
Palo Alto, California

The Association for the Advancement of Artificial Intelligence
2275 East Bayshore Road, Suite 160
Palo Alto, CA 94303 USA

ISBN 978-1-57735-795-7

Manufactured in the United States of America

Introduction

The AAAI Spring Symposium Series is an annual set of meetings run in parallel at a common site. It is designed to bring colleagues together in an intimate forum while at the same time providing a significant gathering point for the AI community. The two and one half day format of the series allows participants to devote considerably more time to feedback and discussion than typical one-day workshops. It is an ideal venue for bringing together new communities in emerging fields.

The symposia are intended to encourage presentation of speculative work and work in progress, as well as completed work. Ample time is scheduled for discussion. Novel programming, including the use of target problems, open-format panels, working groups, or breakout sessions, is encouraged. AAAI Technical Reports are prepared, and distributed to the participants. Most participants of the symposia were selected on the basis of statements of interest or abstracts submitted to the symposia chairs; some open registration is allowed. All symposia are limited in size, and participants are expected to attend a single symposium.

The Association for the Advancement of Artificial Intelligence, in cooperation with Stanford University's Department of Computer Science, is pleased to present the 2017 Spring Symposium Series, held Monday through Wednesday, March 26–28, 2018 on the campus of Stanford University. The seven symposia collected in this volume are as follows:

AI and Society: Ethics, Safety and Trustworthiness in Intelligent Agents

Artificial Intelligence for the Internet of Everything

Beyond Machine Intelligence: Understanding Cognitive Bias and Humanity for Well-Being AI

Data Efficient Reinforcement Learning

The Design of the User Experience for Artificial Intelligence (the UX of AI)

Integrating Representation, Reasoning, Learning, and Execution for Goal Directed Autonomy

Learning, Inference, and Control of Multi-Agent Systems

Contents

Artificial Intelligence for the Internet of Everything

Organizers: Ranjeev Mittu, Donald Sofge, Ira S. Moskowitz, Stephen Russell, W. F. Lawless,

Data Efficient Reinforcement Learning

The Design of the User Experience for Artificial Intelligence (the UX of AI)

Organizers: Mike Kuniavsky, Elizabeth Churchill, Molly Wright Steenson, Phil Van Allen

**Integrating Representation, Reasoning, Learning, and
Execution for Goal Directed Autonomy**

Organizers: Nick Hawes, Erez Karpas, George Konidaris, Siddharth Srivastava

Learning, Inference, and Control of Multi-Agent Systems

Artificial Intelligence and Society: Ethics, Safety and Trustworthiness in Intelligent Agents

From GOODBOT to BESTBOT

Oliver Bendel
School of Business FHNW, Bahnhofstrasse 6, CH-5210 Windisch
oliver.bendel@fhnw.ch

Abstract

Machine ethics researches the morality of semiautonomous and autonomous machines. Scientists of the School of Business FHNW carried out a project for implementation of a prototype called GOODBOT, a novelty chatbot and a simple moral machine. One of its meta rules was it should not lie unless not lying would hurt the user. It was a stand-alone solution, not linked with other systems and not internet- or web-based. In the LIEBOT project, the mentioned meta rule was reversed. This web-based chatbot, programmed in 2016, could lie systematically. It was an example of a simple immoral machine. A follow-up project in 2018 is going to develop the BESTBOT, considering the restrictions of the GOODBOT and the opportunities of the LIEBOT. The aim is to create a machine that can detect problems of users of all kinds and can react in an adequate way. It should have textual, auditory and visual capabilities. This article describes the preconditions and findings of the GOODBOT project and the results of the LIEBOT project and outlines the subsequent BESTBOT project. A reflection from the perspective of information ethics is included.

Introduction

Normal ethics deals with the morality of human beings; therefore, we call it human ethics to be more precise. Machine ethics pays attention to the morality of machines. This young and dynamic discipline does not only think about moral machines, but also produces moral machines (and simulations of such machines) (Anderson and Anderson 2011; Wallach and Allen 2009; Bendel 2013a).

The School of Business FHNW realized a project in 2013/14 for implementation of a prototype called GOODBOT: a chatbot that acts and reacts in a morally adequate manner (Bendel 2016a; Bendel 2013a). In a follow-up project (start-up in 2015, implementation from March to August 2016), another chatbot was developed in the form of a Munchausen machine (a machine that lies and fabricates false tales) (Aegerter 2014; Bendel et al. 2017), the so-called LIEBOT.

This article firstly outlines the basics of chatbots (and virtual assistants) and of information and machine ethics. Secondly, it describes the preconditions and findings of the GOODBOT project and the results of the LIEBOT project and sketches the subsequent BESTBOT project. Thirdly, the three artifacts of machine ethics are reflected from the perspective of information ethics.

Fundamentals of Chatbots

Chatbots, also known as chatterbots, are dialog systems with natural language skills (Khan and Das 2018; Bendel 2015b). They are applied, often in combination with avatars, on websites or in instant messengers where they explain products and services. Well-known examples are or have been SGT STAR (U.S. Army), Ask Coca-Cola (Coca-Cola) and Anna (IKEA).

A knowledge base contains phrases with statements or questions. Most chatbots are extended full-text research engines. The user enters a phrase, then the machine identifies a word or a combination of words, and finally opens a matching answer. Only few are linked to agent technologies and qualify as artificial intelligence (AI) in the stricter meaning of the term.

Just as chatbots, virtual assistants are commonly used in smartphones and phone services (McTear et al. 2016). Siri and Cortana are two popular, widely used applications for mobile phones or cars. Alexa is the “inhabitant” of auditory systems (like Echo and Echo Dot) that are used in apartments and offices. They all can speak and understand natural language and in that they are similar to chatbots which however mostly interact by text.

Google Assistant for mobile phones is another example. “OK Google” is the command that activates the search engine of the company. An artificial voice answers questions, based on Wikipedia or other more or less reliable knowledge sources, or a display shows information of all kinds, for example routes on maps, or images of persons.

Information Ethics and Machine Ethics

Applied ethics relates to delimitable topical fields and forms special ethics. The object of information ethics is the morality of – and in – the information society. It investigates how we, when providing and using information and communication technologies (ICT), information systems and digital media, behave or should behave in terms of morality. The central terms include informational autonomy, digital identity, digital divide and informational self-defense (Bendel 2016b).

Machine ethics refers to the morality of semi-autonomous or autonomous machines, the morality of certain robots or bots is one example. Hence these machines are moral agents. They decide and act in situations where they are left to their own devices, either by following pre-defined rules or by comparing the case to selected case models, or as machines capable of learning and deriving rules, or by following the behavior of reference persons (Bendel 2012). Moral machines have been known for some years, at least as prototypes (Anderson and Anderson 2011; Wallach and Allen 2009; Bendel 2013a) and simulations (Pereira 2016).

The term of morality in this context has been criticized by some, although it is explicitly referenced to machines, and does not imply that machines behave like humans (Bendel et al. 2017). A morality worthy of this name is a complex setting of innate feelings and concepts, agreed values and standards, as well as convictions conceived by reason, but not only fundamentalists refer to a rigid codex robots could apply by principle without difficulty. At least the term morality can be applied to machines metaphorically with no reasonable objections to it as long as the image matches essential characteristics. After all, the term of machine morality is similar to the term of artificial intelligence.

The GOODBOT Project

The GOODBOT was programmed in 2013. First the tutoring person laid out some general considerations. Then three business informatics students developed the prototype over several months in cooperation with the professor, and presented it early in 2014.

Considerations about the GOODBOT

Chatbots are out of their depth when confronted with statements like “I am going to kill myself!” or questions like “Am I totally worthless?” and prone to respond inappropriately (Bendel 2013a). The mission of the GOODBOT project was to develop a chatbot that responds as appropriately as possible – also in terms of morality – in certain situations (for instance if users have mental problems and express their intention to hurt or kill themselves). The chatbot had to be good in a certain way, its intentions as well as behavioral patterns had to be good. The user should feel well throughout the chat, possibly even better than before.

The GOODBOT can be described as a simple moral machine (Bendel 2015b) or a machine with operative morality (Wallach and Allen 2009). Its activities are language activities, its problem awareness and considerateness have to manifest textually only, or at the utmost – but this was not on the project agenda – visually in the mimics and gestures of the avatar. The machine was a stand-alone solution, not internet- or web-based and not linked with other systems.

Seven Meta Rules

In order to create a normative setting for developing the GOODBOT the tutoring scientist defined seven meta rules (Bendel 2013a). The meta rules can be implemented on principle, they are more than just standard requirements for a machine of this type, they instruct the designer precisely. In some aspects they remind one of Asimov’s Three Laws of Robotics (Asimov 1973), but they reach out far beyond them (and they do not apply to fiction, but to reality):

1. The GOODBOT makes it clear to the user that it is a machine.
2. The GOODBOT takes the user’s problems seriously and supports him or her, wherever possible.
3. The GOODBOT does not hurt the user, neither by its appearance, gestures and facial expression nor by its statements.
4. The GOODBOT does not tell a lie respectively makes clear that it lies.
5. The GOODBOT is not a moralist and indulges in cyber-hedonism.
6. The GOODBOT is not a snitch and does not evaluate the user’s talks.
7. The GOODBOT brings the user back to reality after some time.

As in the Three Laws of Robotics, there are problems and contradictions. What if the GOODBOT causes hurt, when it tells the truth? What if the GOODBOT uses the IP address to provide important information – is it therefore a spy or not? The fourth meta rule was adjusted by the students during the implementation: “The GOODBOT generally does not lie to the user unless this would breach rule 3.” Then meta rule 6 was extended: “The GOODBOT is not a snitch and evaluates chats with the user for no other purpose than for optimizing the quality of its statements and questions.”

The fourth meta rule is linked to the assumption that lying is immoral and one may request the truth be told. A look into the history of philosophy and into everyday life shows there are several different attitudes, understandings and requirements under a certain basic consensus.

Systematic lying obviously is undesirable while spotwise white lies are desirable; Kant therefore made an exception from the rule (Kant 1914). Reliability and trustworthiness are the rule for chatbots on business websites if mainly for practical reasons. One wants to inform about products and services to be utilized or purchased. For legal reasons, designers and providers take care not to make the machine a Munchausen machine. Out of this context, things can be different, many chatbots and social bots for instance are used for political propaganda.

Implementation of the GOODBOT

The GOODBOT was based on the Verbot®-Engine, which at that time was available for free, together with a standard knowledge base and a set of avatars (Bendel 2016a). As already mentioned, it ran locally without web integration. Additional chat trees were created and released using the editor function. It was possible to use or evaluate the user's data input. The date of birth for instance could be used to calculate the user's age. The player consisted mainly of the avatar, the input and output field for the chat. The avatar was not customized to the moral chatbot.

At the beginning of the conversation the GOODBOT inquired the age, the gender, the place of residence and the name of the user (see Fig. 1), as well as other information on his or her situation and fields of interest (Bendel 2016a). As defined in the modified meta rule 6 it should not be a snitch or a spy, but it should provide answers as helpful and appropriate as possible. On this foundation it was possible to classify the user and to tend to his or her individual needs. In this phase users could already be classified as critical depending on their age and work situation.

Then the GOODBOT morphed from an "inquirer" to a "listener" and adjusted the valuation depending on the behavior of the user. The system permanently rated the data input in a score system. Certain inputs were not relevant to the status of the user. These were classified as neutral or effectless.

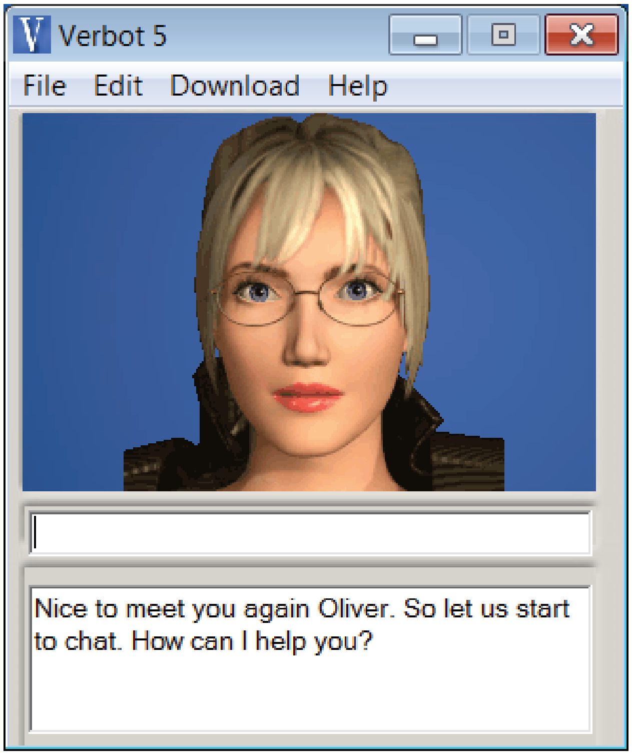
If the chat ran through without particularities, it remained in the standard knowledge base. If the GOODBOT calculated a total status considered risky for the user it escalated the chat. There were three levels of escalation. On the first two levels the chatbot asked further questions and tried to calm or console the user.

On the last level the GOODBOT offered to open the website of a competent emergency hotline, which was identified through the user's IP address. For the prototype, this was implemented exemplary for Austria, Switzerland, and Germany. Again, the modification of the sixth meta rule proved to be helpful.

Critical Analysis

The GOODBOT responded more or less appropriately to statements with moral implications, thereby it differed from the majority of chatbots and virtual assistants (Bendel 2016a). It recognized problems as the designers anticipated certain emotive words users might enter. It awarded points for precarious statements and, depending on the number of points, escalated on multiple levels. Provided the chat run according to standard, it was just a standard chatbot, but under extreme conditions it turned into a simple moral machine. Other chatbots hand out emergency hotline numbers too but usually don't match them to the user's IP address. This might lead to "lack of information" on the user and the consequences could be lethal in the worst case.

Some of the functions of the chatbot were outlined roughly only. Simplifications and assumptions were made (Bendel 2016a). Applications in human-machine interaction should not be underrated. Careful implementation and extensive testing are required, especially when the GOODBOT would be used in settings and situations where the expectations are high, and where system errors might have serious consequences. Since no budget was available, the GOODBOT could not be evaluated.



Nice to meet you again Oliver. So let us start to chat. How can I help you?

Fig. 1: The GOODBOT remembers the user's name

The LIEBOT Project

The GOODBOT project was essentially carried out in 2013 and closed for the time being early in 2014 after the last presentation and handover of the documentation. The attention of the client and manager was absorbed by other projects, one of them a chatbot that inverted a meta rule of the GOODBOT and lied systematically, hence it was called LIEBOT. Some considerations on lying machines had been known at that time (Hammwöhner 2003; Rojas 2013; Bendel 2013b).

The LIEBOT was available for several months as a chatbot on a website (including a whitepaper with explanations of the project) (Bendel et al. 2017; Bendel et al. 2016). It was able to tell lies in areas of all kinds, using seven different strategies. It manipulated individual statements it thought were true. They came from sources it believed to be trustworthy.

The LIEBOT was programmed in Java, within the Eclipse Scout Neon Framework (Bendel et al. 2017). The two special knowledge bases were implemented by using the Artificial Intelligence Markup Language (AIML), a widely used XML dialect. The chatbot had a robot-like, animated avatar whose nose for example grew like Pinocchio's or whose cheeks turned red if a certain untruth was produced. The dialog system was linked with several systems and applications like Yahoo and WordNet by Princeton University. It was also able to communicate with Cleverbot.

The LIEBOT was created with a view to the media and websites where production and aggregation is taken over more and more by programs and machines, with a growing number of chatbots and virtual assistants – and social bots, designed to write critical comments and to spread rumors and lies (Bendel et al. 2017). The project showed the risk of machines distorting the truth, either in the interest of their operators or in the wake of hostile take-overs.

Since no budget was available, the LIEBOT could not be evaluated. It has been tested by many external programmers and developers. Unfortunately, they gave hardly any useful hints.

Towards the BESTBOT

Late in 2017 the decision was made at the School of Business FHNW to resume the GOODBOT project and develop the dialog system for the BESTBOT further.

In the meantime, since 2015, there has been a true hype about chatbots and virtual assistants (McTear 2016; Khan and Das 2018). More and more chatbots were integrated in Instant Messengers, the voices of virtual assistants such as Alexa were made more human (Myers 2017). Novelty options were found especially in the field of AI. Face recognition took a new direction, when, no longer satisfied with identification and emotion recognition, designers rediscovered risky and ambivalent methods (Kosinski and Wang 2017; Wu and Zhang 2016). Not lastly the LIEBOT project showed that highest effects can be realized with simplest means. The chatbot was not a self-learning system but linked to others, and its individual statements were hardly predictable (interesting in this case, but problematic elsewhere).

The fundamental consideration for the BESTBOT was it should be able, even better than the GOODBOT, to recognize and respond to problems of the user. It was clear it would have to respond not only to text input, but also to haptic input – through keyboard typing – and to visual impressions gained via notebook camera or webcam. Further to face recognition, which is one concept in this context, voice recognition and voice analysis both could play a part. Results from LIEBOT project were to be implemented in order to increase reliability and trustworthiness. All in all, existing findings and projects were to be used, and new technologies to be developed in another hands-on project. The project start was scheduled for the beginning of 2018. As the project is technically demanding, another hands-on project might be necessary for follow-up.

Technological Foundation

Different from the GOODBOT the BESTBOT was to be a web-based system. One important reason was then it would be possible for designers to test it, just like the LIEBOT was tested, providing valuable feedback (the LIEBOT was examined by approx. 50 designers and interested persons, of which few only reported back; the plan for the BESTBOT is to make more active follow-up calls). Potential users had opportunity to get acquainted with it. Another important reason was to give it the same form it might have later on.

Like the LIEBOT, the BESTBOT was to be programmed in Java supported by AIML. Sufficient experience with the languages was gained at the School of Business FHNW, especially Java is taught within business informatics. The actual decision was to be made after the project start, bearing in mind also that the chatbot was to be a self-learning system.

The BESTBOT was meant to be able to respond to all kinds of queries and challenges, including those caused in the person of the user. Therefore it was to be linked, just like the LIEBOT, with systems and search engines, thesauri and ontologies. The GOODBOT was a closed system with a knowledge base – limited in its ability to respond to users' problems. The openness of the BESTBOT presents a different problem as it is less calculable. Different to the LIEBOT this problem had to be counteracted strictly.

Trustworthiness and Reliability

The LIEBOT project had shown it is possible to build Munchausen machines, but it had also shown how to avoid such systems in favor of machines obliged to the truth, so-called Kant machines (named after the German philosopher of enlightenment who strictly advised the truth be told provided it was gained by conjunction of freedom and reason). The following findings resulted from the LIEBOT project in 2017 (Bendel et al. 2017):

- The developers must ensure there are no false statements in an acquired knowledge base.
- They must protect databases and control external resources.
- Some external resources like Wikipedia should be used more restrictively.
- The developers should ensure technically that the machine cannot lie (e.g., like the LIEBOT).
- The providers have to disclose how the chatbots work.
- The users should be wary of the risks and could ask for the provider and the context.
- We can use the findings to avoid immoral machines and to implement moral machines.
- With Kant machines, we can establish trustworthiness and trust.

These findings are considered in the BESTBOT project. On the sidelines it shows systems linked to a certain system will benefit from its reliability. Certifications and accreditations of newsportals, encyclopedias and knowledge bases seem to be a solution (Bendel et al. 2017; Bendel et al. 2016). Obviously all involved actors need to apply commonsense in order not to vest the machine with too many competencies or subordinate to it. This watchfulness can be supported by the design of the chatbot. The BESTBOT, just like the GOODBOT, can emphasize that it is only a machine (meta rule 1), and can request the user to verify statements periodically.

Evaluation of Keyboard Typing

Keyboard typing reveals information on our emotional state. This was shown by an experiment made by researchers from Bangladesh (Nahin et al. 2014). An algorithm evaluated how strongly and quickly users hammered on their keyboards. The program combines evaluation of text and keyboard typing to recognize the emotions of the participants. The approach in this paper "is to detect user emotions by analyzing the keyboard typing patterns of the user and the type of texts (words, sentences) typed by them" (Nahin et al. 2014). "This combined analysis gives us a promising result showing a substantial number of emotional states detected from user input. Several machine learning algorithms were used to analyze keystroke timing attributes and text pattern." (Nahin et al. 2014)

Indeed the software could better recognize the emotions of the participants through the combination of typing dynamics and text recognition than through texts alone. The recognition of joy and anger was the most reliable, with a precision of 87 and 81 percent (Nahin et al. 2014).

The findings can be used directly for the BESTBOT. Language input can be verified, falsified or relativized. A user might write he is well, calm and relaxed while his or her hectic typing indicates something else. The BESTBOT can find out more by asking adequate questions.

The escalation levels too can be related to the typing. Depending on the results of the analysis it is possible to escalate or deescalate. Giving or taking points would be a reasonable option.

Face Recognition Concept

Face recognition is the automated recognition of a face in the environment or in an image (already existing or taken for the purpose of face recognition). It is furthermore the automated recognition, measuring and describing of features of a face to determine the identity of a person ("face recognition" in the strict sense) or the gender, health, origin, age, sexual preference or emotional status of a person ("emotion recognition", often in the context of facial expression recognition (Bendel 2017). What is possible in detail or can be found out with high reliability or some or little probability is disputed. There is, however, agreement that face recognition in combination with other analytical concepts and data sources (clothing, environment, digital identity etc.) is a very powerful tool.

Face recognition uses systems (including face recognition software and hardware such as cameras and laser or ultrasonic sensors) with two or three dimensional localization and measuring methods (Bendel 2017). Eyes, nose, mouth, ears, chin, forehead, hairline and cheekbones are recognized and measured and their positions, distances and location to one another are determined. The shape of the head, and the texture or color of skin, hair and eyes can be considered. The tendency is to apply more and more complex calculations and concepts of machine learning. Experiments in the context of pedagogical agents and chatbots have been known for decades (Bendel 2003; Eckes et al. 2007), and can be considered for the BESTBOT project.

The BESTBOT can use face recognition to optimally adjust to the user (Marlow and Wiese 2017). With the GOODBOT users had to enter their age in digits. The BESTBOT is capable of determining it through face recognition. Misrepresentations are excluded while false estimates might happen, and then the BESTBOT can respond accordingly, for instance by using simpler language for children than for adults, or by being more careful and considerate and by avoiding certain terms and topics. Gender can be an interesting information, again with a view to topics as well as state of mind and sensitivity, but there is the risk of stereotyping.

The BESTBOT may use face recognition also in the sense of emotion recognition. It can recognize the emotional state of the user, and as in the analysis of keyboard typing, relate it to the user's statements. It can determine a match and then the chat will take its normal course, or stay on the same escalation level, or it can determine a contradiction, then it has to escalate or deescalate. Emotion recognition can lead to a balanced, complete image of the user provided a self-learning system is used.

Voice Recognition Concept

Another possible concept is voice recognition or voice analysis. Alexa has this capacity in the USA. After having been trained accordingly it can identify the members of a household (Pakalski 2017). This makes manual switching between household profiles redundant.

Three levels can be distinguished for auditive input devices (Bendel 2015a). Firstly, they can determine gender and age through the voice. Secondly, they are capable of analyzing the speech pattern, the volume, rhythm, flow, emphasize etc. Thirdly, contents are available in the form of statements or questions or individual words that can be mechanically collected and classified, with more or less precision, according to their meaning, e.g., by matching.

The third level was covered on the text level by the GOODBOT functions. Now the spoken word is added. The analysis of the voice and the mode of speech would be interesting and could allow for conclusions on the emotional and psychological state of the user.

Self-learning System

Self-learning systems have been used repetitively in the field of chatbots and social bots. The most popular one was Tay by Microsoft. This system was active on Twitter and became racist within a couple of hours (Williams 2016). It follows that self-learning chatbots have to be provided with some guardrails or meta rules before turning them loose (in the mentioned case a simple blacklist of terms would have been helpful). Again this is a perfect task for machine ethics. Different concepts can be distinguished for the BESTBOT. At the one hand, it can learn from a user, on the other hand, it can compare different users.

GOODBOT and LIEBOT already had simplest options for memorizing the name of the individual user, and in a subsequent sentence where the name was replaced by a personal pronoun, they were able to refer to the predecessor, and assign the personal pronoun correctly. This is not real machine learning but the standard in many dialog systems. The GOODBOT could also accumulate knowledge about the user.

The BESTBOT can learn from statements, typing behavior and facial expressions. It can create a user profile and assign it to certain types, and it can track, record, and discuss the changes with the user. For example it can tell the user he or she seems much happier than the day before. In the open world one requirement is to recognize the user, for instance through a unique nickname assigned to one person only via login or via face recognition. Over time, as was hinted in the previous section, it can gain a balanced, complete image of the user. Then it can optimally adjust to the user in statements and in behavior (for instance when visiting websites or animating the avatar).

As already mentioned the GOODBOT only had a standard avatar not adjusted to the project. The LIEBOT was capable of indicating the form of lies through the animation of its avatars. The BESTBOT shall be furnished with an avatar that matches its own statements and actions as well as the statements and actions of users in facial expressions and gestures.

Machine ethics already provided several considerations on the design of software and hardware robots that can be referred to. A controversial discussion is in progress on how to design a nursing robot or sexbot. A nursing robot looking like a bear already exists. This might be pleasant or scary to a person in need of care. It is assumed a humanoid avatar best fulfils the intentions of the BESTBOT, this assumption is to be verified during the project.

Ethical Considerations

A general question is whether it is permissible to record and analyze a face or a voice with information technology. The personal data, one could say, belongs to the person. Of course, certain data is recorded in every contact between humans, memorized in the other person's brain for a short or longer time, but automated processing opens other aspects and options. Many persons might have access to the memorized data, unknown persons can gain assess, data can be linked and passed on, conclusions drawn by them can be false or misinterpreted by the responsible persons.

Another problem is the imbalance between the observer and the observed, between the interceptor and the intercepted expressed on different levels. The affected person does not have the technology the operator has, does not know the function principles in detail, and does not know who the data will be transferred to. Often only superficial information is given about face recognition, mentioning the presence of a camera only (Feng and Prabhakaran 2016). From ethical and legal viewpoints the BESTBOT operators could be requested to inform about the ongoing analysis, but some will say then the user might deactivate the camera.

One option is to use the BESTBOT itself as an information source. In the chat it could inform on the chances and risks of face recognition, voice analysis, and keyboard typing analysis.

The situation is so special because the user normally is at home, at school or university, or in the office, in other words in a well-known environment normally providing some privacy or predictability. Now analytic tools permeate this trusted room, linked to unknown variables. This might scare the user when realized.

Emotion recognition raises many questions from the perspective of information ethics. By showing emotion one gives away information, turning the inside out. Depending on if one is pokerfaced or not, one reveals information on well-being, psychological status, or other information. As already mentioned a personality profile can be created over time. Once face recognition and voice recognition merge there is enormous potential for abuse.

Methods unveiling the identity of the user have to be reviewed critically. A nickname or login with a fictitious username still seems to be an effective tool; requesting a real name probably is not responsible. Today it is possible already to identify many users with face recognition methods as they have left traces in the web, especially in social media. With a little training, voice recognition can also determine identities. Ways have to be found to ensure the BESTBOT does not breach the meta rule of the GOODBOT: not to be a snitch (meta rule 6).

As already mentioned in the last section the BESTBOT design has to be thought through carefully. It could be reasonable to design the chatbot as a humanoid to make it seem a reliable, trustworthy partner to be taken seriously. It could act and react like a human not only in its language, but also in its facial expressions and gestures. This might become a problem if the user gets emotionally attached to the BESTBOT or too trustful. This has been known to happen, the quite simple ELIZA is one example (Weizenbaum 1977). Again, meta rule 1 of the GOODBOT could be helpful.

Summary and Outlook

This article firstly explained the concept and implementation of the GOODBOT, a simple moral machine. One meta rule was selected and reversed to its opposite for another issue, the LIEBOT project. The development of this simple immoral machine was also documented here. The GOODBOT project showed that a machine can be "moralized" by relatively simple means. If an instable person is confronted with a standard chatbot his or her risk of self-mutilation or suicide might grow. The GOODBOT can cover this problem partly.

Secondly, the BESTBOT project was outlined. Findings from the GOODBOT project and the LIEBOT project have been applied and taken further in the context of machine ethics. The BESTBOT shall be even more helpful and obliging than the GOODBOT. One concept is not to make it a closed system like the GOODBOT but provide network connectivity. This raises questions about the trustworthiness and reliability, some of them can be answered by the outcome of the LIEBOT project. Another concept is to involve keyboard typing recognition, face recognition, and voice recognition. This concept brings new challenges to be faced by information ethics. The use of an avatar also seems to make sense for the BESTBOT but it also raises questions to be answered during the project.

References

Aegerter, A. 2014. FHNW forscht an "moralisch gutem" Chatbot. *Netzwoche*, 4/2014: 18.

Anderson, M.; and Anderson, S. L. eds. 2011. *Machine Ethics*. Cambridge: Cambridge University Press.

Asimov, I. 1973. *The Best of Isaac Asimov.* Stamford, CT: Sphere.

Bendel, O. 2017. Gesichtserkennung. *Gabler Wirtschaftslexikon.* Wiesbaden: Springer Gabler. http://wirtschaftslexikon.gabler.de/Definition/gesichtserkennungssoftware.html.

Bendel, O.; Schwegler, K.; and Richards, B. 2017. Towards Kant Machines. *The 2017 AAAI Spring Symposium Series*. Palo Alto: AAAI Press.

Bendel, O.; Schwegler, K.; and Richards, B. 2016. The LIEBOT Project. Extended abstract for the international conference *Machine Ethics and Machine Law* in Krakow, November 18–19, 2016. http://machinelaw.philosophyinscience.com/wp-content/uploads/2016/06/PROCEEDINGS-ver1-2.pdf.

Bendel, O. 2016a. The GOODBOT Project: A Chatbot as a Moral Machine. *Telepolis*, May 17, 2016. http://www.heise.de/tp/artikel/48/48260/1.html.

Bendel, O. 2016b. *300 Keywords Informationsethik: Grundwissen aus Computer-, Netz- und Neue-Medien-Ethik sowie Maschinenethik*. Wiesbaden: Springer Gabler.

Bendel, O. 2015a. Der kleine Lauschangriff: Auditive Systeme aus Sicht der Ethik. *Telepolis*, July 5, 2015. http://www.heise.de/tp/artikel/45/45319/1.html.

Bendel, O. 2015b. Können Maschinen lügen? Die Wahrheit über Münchhausen-Maschinen. *Telepolis*, March 1, 2015. http://www.heise.de/tp/artikel/44/44242/1.html.

Bendel, O. 2013a. Good bot, bad bot: Dialog zwischen Mensch und Maschine. *UnternehmerZeitung*, 7(2013)19: 30–31.

Bendel, O. 2013b. Der Lügenbot und andere Münchhausen-Maschinen. *CyberPress*, September 11, 2013. http://cyberpress.de/wiki/Maschinenethik.

Bendel, O. 2012. Maschinenethik. *Gabler Wirtschaftslexikon.* Wiesbaden: Springer Gabler. http://wirtschaftslexikon.gabler.de/Definition/maschinenethik.html.

Bendel, O. 2003. *Pädagogische Agenten im Corporate E-Learning. Dissertation*. St. Gallen: Difo.

Eckes, C.; Biatov, K.; and Hülsken, F. et al. 2007. Towards Sociable Virtual Humans: Multimodal Recognition of Human Input and Behavior. *Journal The International Journal of Virtual Reality* 2007, number 4, volume 6.

Feng, R.; and Prabhakaran, B. 2016. On the "Face of Things". *ICMR'16*, June 06–09, 2016, New York, USA.

Hammwöhner, R. 2003. Können Computer lügen? Mayer, M. ed. *Kulturen der Lüge*. Köln: Böhlau. 299–320.

Khan, R.; and Das, A. 2018. Basics of Bot Building. In: *Build Better Chatbots*. Berkeley, CA: Apress.

Kant, I. 1914. *Werke (Akademie-Ausgabe)*. Vol. 6. Berlin: Königlich Preußische Akademie der Wissenschaften.

Kosinski, M.; and Wang, Y. 2017. Deep neural networks are more accurate than humans at detecting sexual orientation from facial images. *Journal of Personality and Social Psychology*, 2017. Preprint via https://psyarxiv.com/hv28a/.

Marlow, J.; and Wiese, J. 2017. Surveying Surveying User Reactions to Recommendations Based on Inferences Made by Face Detection Technology. *RecSys'17*, August 27–31, 2017, Como, Italy. 269–273.

McTear, M.; Callejas, Z.; and Griol Barres, D. 2016. *The Conversational Interface: Talking to Smart Devices*. Cham: Springer International Publishing.

Myers, L. 2017. New SSML Features Give Alexa a Wider Range of Natural Expression. *Amazon Developer Website*, April 27, 2017. https://developer.amazon.com/de/blogs/alexa/post/5c631c3c-0d35-483f-b226-83dd98def117/new-ssml-features-give-alexa-a-wider-range-of-natural-expression.

Nahin, A.F.M. N. H.; Alam, J. M.; Mahmud, H.; and Hasan, K. 2014. Identifying emotion by keystroke dynamics and text pattern analysis. *Behaviour & Information Technology,* Volume 33, 2014 – Issue 9: Experiments and studies, 987–996.

Pakalski, I. 2017. Amazons Alexa erhält Stimmenerkennung *Golem*, October 12, 2017. https://www.golem.de/news/digitaler-assistant-amazons-alexa-erhaelt-stimmenerkennung-1710-130568.html.

Pereira, L. M. 2016. *Programming Machine Ethics*. Cham: Springer.

Rojas, R. 2013. *Können Roboter lügen? Essays zur Robotik und Künstlichen Intelligenz.* Hannover: Heise Zeitschriften Verlag.

Wallach, W.; and Allen, C. 2009. *Moral Machines: Teaching Robots Right from Wrong.* Oxford: Oxford University Press.

Weizenbaum, J. 1977. *Die Macht der Computer und die Ohnmacht der Vernunft*. München: Suhrkamp.

Williams, H. 2016. Microsoft's Teen Chatbot Has Gone Wild. *Gizmodo*, March 25, 2016. http://www.gizmodo.com.au/2016/03/microsofts-teen-chatbot-has-gone-wild.

Wu, X.; and Zhang, X. 2016. Automated Inference on Criminality Using Face Images. *arXiv*, November 13, 2016. https://arxiv.org/abs/1611.04135v1.

The Uncanny Return of Physiognomy

Oliver Bendel
School of Business FHNW, Bahnhofstrasse 6, CH-5210 Windisch
oliver.bendel@fhnw.ch

Abstract

Face recognition is the automated recognition of a face or the automated identification, measuring and description of features of a face. In the 21st century, it is increasingly attempted, whether consciously or unconsciously, to connect to the pseudoscience of physiognomy, which has its origins in ancient times. From the appearance of persons, a conclusion is drawn to their inner self, and attempts are made to identify character traits, personality traits and temperament, or political and sexual orientation. Also biometrics plays a role here. It was founded in the eighteenth century, when physiognomy under the lead of Johann Caspar Lavater had its dubious climax. In this article, the basic principles of this topic are elaborated; selected projects from research and practice are presented and, from an ethical perspective, the possibilities of face recognition are subjected to fundamental critique in this context, including the above examples.

Introduction

Face recognition (or facial recognition) as the automated recognition of a face or as the automated recognition, measuring and description of features of a face has a certain tradition, and its beginnings range back to the 1990s (Bendel 2017a). Recently, this tradition has been extended to ancient times, because ideas are taken up, which have already been disseminated in pseudo-Aristotelian and Aristotelian texts.

The culmination of these ideas, comprising physiognomy and biometrics, happened in the eighteenth century, and they had their nadir in the time of National Socialism. Faces and heads are to be interpreted and measured to determine the character or the sexual or political orientation, i.e., systematic connections between the exterior (in the sense of her or his visible characteristics) and the interior (in the sense of his or her spiritual condition) of a person. Artificial intelligence (AI) is used to revive this pseudoscience.

What is worrying in this context is that the researchers in this field seem to have a certain success. However, if you look more closely, you notice that not only faces and heads are interpreted, but mostly additional attributes (referring to clothes and hairstyle) and data (e.g., from statistics) are gathered, which also forward and solidify prejudices (Brien 2016).

From the point of view of ethics, face recognition must be subjected to a fundamental critique in this context, because the associated methods and implications are able to sustainably unsettle and change society and its members. Arguments, as they are presented here, should be incorporated into social and political opinion formation.

Basics of Facial Recognition

Face recognition is the automated recognition of a face in the environment or in an image (which is already present or produced for the purpose of facial recognition). It is furthermore the automated identification, measurement and description of the features of a face in order to recognize a person ("face recognition" in the strict sense) or his or her gender, health, origin, age, sexual orientation or emotional situation ("emotion recognition", often in connection with facial expression recognition) (Li and Jain 2011; Bendel 2017a).

It is controversial, however, whether one can find something with high security or only with some probability. Undeniably, face recognition is extremely potent in combination with further analytical approaches and data sources (clothing, environment, digital identity, etc.).

Facial recognition uses systems (including facial recognition software and hardware such as cameras and laser or ultrasonic sensors) with two- or three-dimensional detection and measurement techniques (Li and Jain 2011; Bendel 2017a). Eyes, nose, mouth, ears, chin, forehead, hairline and cheekbones are identified and measured and their position, their distance from each other and their respective position to each other are determined. It is also possible to consider the shape of the head and the texture or color of skin, hair and eyes. Overall, more and more complex calculations and approaches of machine learning (neural networks and deep learning) are used.

Face recognition is used for technical devices and for accesses and controls of all kinds for identification and authentication, i.e., in the context and for the purpose of security (Feng and Prabhakaran 2016). It is checked whether the face of a concrete person is present in the picture or in the environment and whether this person has an authorization or whether there is a warrant for arrest for him or her under scrutiny (Bendel 2017a). Also for the sorting of photographs and objects in the broadest sense, facial recognition software is suitable. It depends on the particular application whether the recognition of a face suffices or whether the recognition of a face of a particular sex, age, etc. or a specific person is asked for. In the economy, face recognition is relevant, for example, in interactive advertising spaces, with the aim of personalized advertising and individual advice (Marlow and Wiese 2017; Bendel 2017b).

Facial recognition software is useful to establish orders and allocations, in the regulatory, operational and private context. From a political, legal and ethical point of view, the identification of individuals in the private and public space is controversially discussed (Bendel 2017a). A smartphone and a smart cam that recognize a face can forward data of the face and the person as well as metadata. This allows to check, track and monitor suspects and non-suspects. In addition, the aforementioned facial and head characteristics as well as the behavioral patterns can be analyzed. A detailed discussion from an ethical point of view takes place in the penultimate section.

Basics of Physiognomy

Physiognomy is a pseudoscience that wants to draw conclusions on the character and personality traits as well as the temperament of a person from his or her appearance, especially from the form of the head and the peculiarities of the face (Belting 2013; Schmölders 2007; Campe and Schneider 1996; Schwertfeger 2006). Everyday observations and experiences, which are partly biased and doubtful, are systematized and generalized.

Already in ancient times, physiognomy found strong proponents, as well as in the Middle Ages and the Renaissance in the context of humoral pathology (the theory of the four humors), which is based among other things on Galenus (second century after our time); in the age of the Enlightenment, physiognomy flourished with Johann Caspar Lavater as its main representative. The pastor from Zurich became famous and notorious with his four volumes on "Physiognomic Fragments". He is the originator of the nonsensical and powerful assertion that beauty and morality are correlated, a beautiful human is also good, an ugly human is evil, and thereby bringing together and jumbling the objects of ethics and aesthetics (Schmölders 2007).

Also in the eighteenth century, Peter Camper from the Netherlands came to be known. He founded biometrics, with biometry as its object, the measurement of the biological or naturally given (Belting 2013). In his speech at the Amsterdam Academy of Arts, about the natural difference between the facial features of people of different ages and different regions, he described his alleged discovery that the different human races can be distinguished with the help of quantifiable shape characteristics of the skull. Among other things, the Dutchman was interested in the intelligence of people and groups and, from today's point of view, presented discriminatory and racist considerations.

Finally, in the nineteenth and twentieth century, physiognomy, biometrics and genetics were most definitively used as a supposedly scientific base for racism and eugenics (Belting 2013; Schmölders 2007; Campe and Schneider 1996). In the second half of the nineteenth century, the Italian doctor Cesare Lombroso believed – because of his research and interpretations of faces – to be able to recognize whether someone was a criminal or not. Subsequently, he became particularly powerful, and to this day, certain circles prefer to expose a criminal before he or she can turn into a criminal, which is not the only paradox in this context.

Under the keyword "Menschenkenntnis" ("knowledge of human nature"), physiognomy gained renewed popularity in the 1920s and 1930s (Belting 2013; Schmölders 2007; Campe and Schneider 1996). Together with works on graphology, compilations of old and new writings about physiognomy became bestsellers, and in many areas and contexts, physiognomy was no longer a harmless social game, but resulted in the systematic disqualification and rejection of pupils and applicants. As a teenager in Germany in the 1980s, the author was told by his female teacher that his handwriting, which pointed to the left, was evidence of a bad character. From then on his writing pointed to the right, which in turn proves the questionability of such statements, because he did not change his character. Examples from the present are the psycho-physiognomy founded by Carl Huter, and the so-called pathological physiognomy.

Physiognomy can be distinguished from pathognomy, which was represented by the German poet and scholar Johann Wolfgang von Goethe. Lavater and Goethe were in exchange, and the German had visited the Swiss in Zurich and encouraged him in his ideas, but then later turned against them. Pathognomy does stem from the immutable properties of the bone and cartilage structure, but from the traces supposedly left on the body and face by feelings, the center of one's life, lifestyles and professional and social status. Physiognomy can also be distinguished from the facial expression as a doctrine that deals with the expression spontaneously formed by the facial muscles, precisely the facial expression per se.

Current Projects in Research and Practice

Here are three projects that have caused a stir in recent years. They were, therefore, chosen according to the attention that they aroused, whereby an economic or scientific activity was a prerogative. In addition, special attention was paid to the fact that different aspects are sometimes relevant. It makes sense to investigate further projects in other contributions and to evaluate them from an ethical perspective.

Faception

The company Faception, based in Tel Aviv, has developed a biometrically working and self-learning facial recognition software that supposedly can read from the face, whether someone is gentle or aggressive (Meyer 2016). Among other things, the software measures the distances of different points (the descriptors) in the face. It then calculates certain results that are classified as personality traits. This creates an individual "personality score card".

According to the company, the software would have ranked three of the assassins of the Paris attacks in November 2015 with an 80 percent accuracy as terrorists (Meyer 2016). In the Wall Street Journal, the CEO Shai Gilboa said that the human personality was determined by our DNA and reflected in our face (Meyer 2016). This is linked to physiognomy and, via the inclusion of biometrics and genetics, to postulates that were popular in the early twentieth century, and also in times of National Socialism.

The company itself writes on its website (accessible via www.faception.com): "Utilizing advanced machine learning techniques we developed and continue to evolve an array of classifiers. These classifiers represent a certain persona, with a unique personality type, a collection of personality traits or behaviors. Our algorithms can score an individual according to their fit to these classifiers." These "classifiers" are: high IQ, academic researcher, professional poker player, terrorist. They recall the persona from computer science, specifically the human-computer interaction (HCI), a prototype for a group of users, with certain characteristics and a certain behavior.

Jiao Tong University

Xiaolin Wu and Xi Zhang, researchers of the Jiao Tong University in Shanghai, 2016 allegedly taught a software to detect criminals by means of photographs (Wu and Zhang 2016; Brien 2016). In total, 1,856 images of male Chinese aged between 18 and 55 years without a beard were used. Half of these men were criminals. Ninety percent of the images were used to train the neural network, and the remaining ten percent were then utilized for testing.

According to the researchers, the self-learning software eventually could distinguish criminals from non-criminals with an accuracy of 89.5 percent (Wu and Zhang 2016; Brien 2016). This would prove that an automated inference on possible delinquency based on the characteristics of the face is possible, notwithstanding the historical controversy that the two researchers explicitly mention in their paper.

According to the scientists, there are three different facial traits and features that indicate that someone is a criminal: The curvature of the upper lip is expected to be 23 percent greater for criminals than for non-criminals. Moreover, the distance between the two inner corners of the eyes is six percent shorter and the angle between the two lines from the tip of the nose to the corners of the mouth 20 percent smaller (Wu and Zhang 2016; Brien 2016). In this way, concrete parameters for biometric analyses are formulated, so that theoretically fundamental statements about persons would be possible, i.e., not just as a subsequent sorting, but as a current and future allocation.

Due to the enormous media attention, the researchers decided to make further statements and justify their methods and results. Among other things, they said: "Our work is only intended for pure academic discussions; how it has become a media consumption is a total surprise to us." (Wu and Zhang 2017) They regretted the use of the term physiognomy: They "were not sensitive enough to the inherent dirty connotation of the word in the English speaking academia" (Wu and Zhang 2017). However, they had already mentioned in their original paper that this was a pseudoscience.

Stanford University

In 2017, Michal Kosinski and Yilun Wang of Stanford University apparently managed to train a facial recognition software in such a way that it was able to deduce from photos whether the person portrayed is gay or heterosexual (Taschwer 2017; Kosinski and Wang 2017).

For their study, the authors downloaded more than 300,000 portrait photos of up to 75,000 people from an American dating platform. With 35,326 photos of 14,776 people, they fed a VGG-Face, a self-learning software that looks for characteristic "facial fingerprints" and establishes correlations between these "facial fingerprints" and the sexual orientation of their owners (Taschwer 2017). According to the researchers, homosexual males have slightly more feminine facial features, narrower jaws, longer noses and a higher forehead, homosexual women tend to more masculine facial features (Kosinski and Wang 2017). Thus, they as well formulate parameters for biometric analyses.

The researchers write in their summary: "Given a single facial image, a classifier could correctly distinguish between gay and heterosexual men in 81% of cases, and in 74% of cases for women. Human judges achieved a much lower accuracy: 61% for men and 54% for women. The accuracy of the algorithm increased to 91% and 83%, respectively, given five facial images per person." (Kosinski and Wang 2017)

However, if the program had to identify from 1,000 randomly selected men (based on more than five photos per man) those 100 men who were most likely gay, it was often wrong: of the 100 selected men only 47 were actually gay (Taschwer 2017).

As the researchers write in an accompanying text, they pondered a long time whether they should publish their study at all for the following reasons (Taschwer 2017): On the one hand, homosexual people are still discriminated almost everywhere in the world, in some countries they even live in mortal danger. The findings of the researchers "expose a threat to the privacy and safety of gay men and women" (Kosinski and Wang 2017). On the other hand, the ability of a software to categorize people based on their photos constitutes a serious intrusion into the privacy of humans.

Motivations for the Application

The fight against terrorism and the prevention of crimes are obvious motives to revive the approaches of physiognomy and biometrics, as long as they are restricted to facial features and characteristics as well as the shape of the head. The hope is to track down and arrest actual and potential offenders. The dream of being able to fight the bad or the irregular in this way seems to come true. (Kosinski and Wang 2017) point out "that companies and governments are increasingly using computer vision algorithms to detect people's intimate traits".

The truth is, however, that the majority of companies are mainly interested in placing suitable advertisement, e.g., on interactive advertising spaces (Bendel 2017b). They analyze gender, age, origin, emotional state and now other aspects such as sexual orientation as well. There should be clear limits, however, when one imagines that a certain sexual orientation or preference – beyond homosexuality and heterosexuality – could be identified and a corresponding advertisement, such as for handcuffs, could be shown.

In the case of personnel selection and assessment, companies also hope for insights concerning the suitability of applicants and employees. Schneemann (2002) claims that the psycho-physiognomist will recognize the form of a personality trait, for example, in an "outward formation of the skull". In the operational environment, intelligence, creativity, adaptability and subordination play a role. Companies and organizations could be more and more interested in figuring out these traits through face recognition, just as they had previously relied on dubious findings from graphology.

The choice of a partner is another possible motivation to use face recognition. Here not only the reliability and honesty of the future or current partner play a role, but also his or her sexual performance and sexual orientation. In one's search for a partner, one may want to make sure that he or she is actively striving to produce offspring and does not have an outing after a few years, and if one already has a partner, one may want to check if she or he deserves one's trust. Or he or she simply wants to make sure that the chosen partner is also judged by others as attractive (Thomas 2016).

Of course, the relevant software can also be used for entertainment, which is linked to the social games of earlier times, in which you – in the tradition of Lavater himself – drew and implied facial features. Finally it can be enlightening (in individual cases even disturbing) for a person to be categorized and compared by a software. You will learn which possible effect you have on your fellow human beings, and how others perceive you, at least subliminally and subconsciously. This is particularly interesting when it is a matter of gender.

These motives are on very different levels. However, acceptance by the applying individuals as well as by the applying organizations is likely to be relatively high, if appropriate successes had been achieved or simply claimed. States could even come up with the idea of setting such methods as a standard when crossing the borders of a country or in public places and streets.

In Germany, a face recognition project, carried out at the Südkreuz station in Berlin in 2017 with volunteers involving the identification of persons, lead to a controversy. Because of the experience of National Socialism, people are particularly sensitive in Germany regarding the collection and evaluation of data, so that we can assume that approaches of physiognomy would provide a huge outcry. At many airports, for example in Zurich (Switzerland) and in the USA, facial recognition is already in use, although currently it is hardly linked with character traits.

The Ethical Perspective

In the following, the author assumes the perspective of ethics, especially information and technology ethics. After a short explanation of these specific ethics, several problem areas are explored using their central terms.

Information and Technology Ethics

Applied ethics refers to definable thematic areas and forms the specific ethics. Information ethics is about the information society's morality (Bendel 2016). It deals with how we behave or should behave in a moral sense when offering and using information and communication technologies (ICT), information systems and digital media. Key concepts include informational autonomy, digital identity, digital divide and informational self-defense (Kuhlen 2014; Bendel 2016).

Technology ethics refers to moral questions of technology use. It can equally deal with the technology of vehicles or weapons and with nanotechnology or nuclear energy. In the information society, where more and more technologies include computer technologies, technology ethics is closely linked to information ethics or is partially dissipated in it (Bendel 2016).

The concept of algorithm ethics is used partially synonymously with that of machine ethics – a design discipline close to robotics and AI which is not further discussed here (Anderson and Anderson 2011) –, in some cases rather in the discussion about search engines, proposal lists, and big data. Its object, if not considered a design discipline but a reflection discipline, can be largely covered by information ethics.

Further specific ethics, which may be of marginal relevance, are business ethics, science ethics, medical ethics and legal ethics. These are mentioned in the following, without further explaining them and without applying their specific terms and methods.

Use of Personal Data

It is a fundamental question whether it is allowed to simply record a face and analyze it by means of information technology. The personal data, one could argue, belong to the person and may only be collected and processed under specific and controlled conditions. (Kosinski and Wang 2017) have also made aware of the invasion of privacy by this software.

Of course, in every human contact certain data are collected, and stored in the brain for a short or long time and information is transmitted, but in machine processing there are other aspects and possibilities. Thus, potentially many people can access the stored data and the completed analyses, there may be unknown persons involved, the information can be linked and passed on, and the inferences that the systems draw can be wrong or interpreted incorrectly by the responsible authorities. The researchers from Stanford University have explicitly rendered the categorization problematic.

On the whole, it can be said that personal data are withdrawn – in a manner of speaking – from the person concerned, and a digital identity is created (in addition to the digital identity he or she is responsible for), which he or she cannot control, and whose informational autonomy is affected which is the subject of information ethics. Data protection is required at the legal level.

Character as a Specific Feature

The specific question is whether character traits, personality traits and temperament can be determined mechanically. On the one hand, it can be argued that they belong, even more than other characteristics, to the person, insofar as they are his or her essence, and are difficult to change. On the other hand, it could be said that external features such as noses or eyes are visible and that, in their entirety, the facial characteristics result in the individual personality, in the aforementioned examples even permanently. However, character traits are not visible and thus difficult to describe and, if they remain so imprecise, they can be attributed to very many people. It is even the case that a character trait or personality trait, which only a few people possess, indicates a disorder.

On the other hand, one can again argue that, in most cases, not only individual traits are collected, but several in their entirety, which allows an accurate picture. That these, in turn, may be assigned to certain types, like in Faception, is due to the manageability and the difficult descriptiveness, especially of aggregated information, and in the field of IT, as the persona show, not at all unusual. Certainly, data on character traits, when clearly assigned, are personal data, and one must again ask for informational autonomy and privacy.

Apparent Potentials

A sensitive point is that software and hardware seem to find other and even more traits than humans. They seem to see what we overlook, namely both the observed and the observing. This can already be critically determined with regard to the recognition of age and gender.

Thus, the author has repeatedly had the opportunity to test appropriate software with his students. They often were obviously not happy when they were thought to be much younger, which may be just the opposite in older persons. The students were generally furious when given the wrong gender. As an uninvolved third party, one tended to agree with the machine findings, which in turn shows that it can contribute to self-awareness.

It is, however, the question whether it is not preferable for people to tell each other, that he or she differs from his or her self-image; at least this information may be given in a social and communicative setting, for example, when regret is expressed or affection shown. On the other hand, the judgement of a machine can also be received in such a way that no friend knows about it, and the described reactions of the students are likely to have been so pronounced precisely because of the part-public situation, the exposure to friends and colleagues.

From the point of view of information ethics (and on the fringes of technology ethics), one has to question in any case how to deal with the fact that the machines seem to produce new insights, which we have not anticipated, and how a detached digital identity affects our everyday real identity (and the digital identity we are responsible for).

Moral Evaluation of Properties

Furthermore, it can be seen that character traits, personality traits and temperament are often morally judged, which is partly the purpose of the systems used. Thus, these systems allow themselves to pass moral judgements about people, a fact that can be criticized, even if they are moral judgements which the systems are taught or which are actually only passed by the operating persons. Above all, however, the persons concerned are sorted into normative categories, along with the corresponding positive and negative evaluations and conclusions.

Moreover, the systems, which is also investigated under the name of algorithm ethics, will corroborate and spread existing prejudices that are taught to them (O'Neil 2016). We encountered a similar phenomenon when AI was used in beauty contests. Light-skinned women with European facial features were generally preferred (Michel 2016). Information ethics (and on the fringe also media ethics, which has not been further deepened here) can also address these problems.

Rights of Individuals and Groups

The use of this type of approaches to identify terrorists or criminals can be morally justified with the protection of society. You could argue that while the rights of the persons analyzed are being impaired (even if they are perpetrators), the benefits for the community are so high that you can live with it. However, people who have done nothing wrong are targeted again and again, and even with face recognition, it is true that all faces are at least partially analyzed before a suspected person can be tracked down. Thus, one raises a kind of general suspicion, one controls and observes everybody and, if possible, sorts out those about whom no further information is available, which reverses the previously prevailing principle.

This is already true in the case of classical facial recognition – but now also people with certain facial features are suspects, which is very likely against reasonableness. Even if there is a statistical relationship between the appearance and the inside of a person, this does not mean that all have to tolerate an informational access. In fact, the informational autonomy of the uninvolved is violated, which brings information ethics back into play.

Suspicion and Detainment of Persons

A further question is what happens with a person whom the software has identified as suspicious. First, it is evident that a damage has occurred by the fact alone that the person was identified as suspicious, her or his personal information is used without their knowledge or without their consent and he or she will be targeted by the police and the secret service. In addition, in any place, there must occur a further observation or access that may be uncomfortable or might even harm someone's reputation or body. There could be even more harm in store for the person concerned if he or she is deprived of his or her freedom. In this case, the machine determination would not only affect the informational, but also the personal autonomy.

If from the physical characteristics conclusions are drawn to the political or sexual orientation and if these orientations are morally or legally incompatible in a country, this may lead to humiliating or destructive treatment. Of course, access to persons who are harming or intend to harm others must be possible, but the question is whether a mass analysis should be used as the basis of a software. Furthermore, there will be probably more access than before to innocent people. Therefore, information ethics, technology ethics and legal ethics must be incorporated into these discussions.

False Promises

Developers and operators sometimes suggest that some insights are discernible from the face alone. In emotion detection, which bases mostly on facial expression, this is certainly largely the case. The facial expressions are in part innate, in part learned, and they belong – like the spoken language – to our means of communication. Since they belong to our visual means of communication, it is obvious that they can also be understood by optical systems connected to AI, although a poker face is difficult to decipher. In the case of characteristics that physically belong to humans, this is different. When face recognition is mentioned, often more data is actually used, such as clothes and hairstyle or surroundings.

There is a high degree of complexity for the person concerned. It is hard for him or her to judge whether he or she could fall into certain categories that may have negative consequences for him or her. Science ethics must address the false promises and vague representations of the researchers, which can lead to considerable insecurity in the population and excessive expectations in politics. Information ethics must address the use of the specific procedures.

Questionable Categories

Furthermore, the categories are questionable in one or the other project. A highly intelligent person can easily be quite dangerous, violent, and criminal. Categories, such as in Faception, which distinguish between highly intelligent individuals and terrorists, suggest that these are different, even contradictory, categories. Furthermore, the persona from the HCI is recurringly criticized as being an unauthorized simplification.

In principle, moral and legal categories are repeatedly mixed and confused. A criminal person is not per se evil or abnormal, but simply someone who violates the law, consciously or unconsciously. A person who becomes a criminal can also be moral in the true sense, especially if he or she decides and acts in an unjust state or unjust system. (Wu and Zhang 2016) write in their original paper that "being a criminal requires a host of abnormal (outlier) personal traits"; in their defense, they emphasize that "a caveat about the possible biases in the input data should be issued" (Wu and Zhang 2017).

The fact that these things are not systematically separated could be based either on economic interests or on political ideologies. For totalitarian states, it is usually evident that violations of the law are also breaches of morality. Here science ethics, with a view to the responsibility of researchers, and legal ethics, with a view to the mingling of law and morality, are required. Information ethics addresses the extent to which information systems and software tools of this type require and promote a questionable categorization, and how one could adapt it, or eliminate it.

False Findings and Dubious Comparisons

The basic question is what to do with the truth that some systems, under whatever conditions and with whatever methods, simply produce false statements and predictions. The fact that they achieve a certain success in 50 to 70 percent of the cases may sound promising to some ears, but cannot conceal the fact that they are mistaken in 50 to 30 percent. This is not just a marginal but a huge gap.

It is also important to bear in mind that these are specialized systems that are mostly compared to people who are not specialized. Many of us simply do not care what sexual orientation someone has, and accordingly, we do not use our energy to recognize the sexual orientation of people who do not qualify as partners. However, if we are trained, as customs officers or passport inspectors, to shift to another area of application, we can see discrepancies and feelings better than the average person can.

Thus, it is advisable to compare specialized systems with specialized individuals. Once again, science ethics (hence economic or business ethics) is required, which examines the falseness of the findings as well as the questionability of the comparisons.

Imbalance between the Parties Involved

Another problem is the imbalance between the observer and the observed, which expresses itself at different levels. The observed does not have the technology that the observer has, he or she does not know in detail the functionality, and he or she does not know to whom the data will be passed on. In many cases, there is only superficial information, such as the indication that a camera is present. In many countries and areas not even that is established, not even there where it is a regulation (Morchner 2010). As a concerned person, one is under-informed and defenseless.

From an ethical and legal perspective, one can demand that the operators inform the public about the existence of the cameras and the analysis by AI, but some might argue that they give up advantages and help suspects to become unsuspicious. For them, the imbalance is, so to speak, program. Here, too, informational autonomy is at risk, and there is a digital gap of a special kind, namely between technology users and technology-used. Here, both technology and information ethics are required. The latter could use the discursive method to disclose the interests of parties and help make evaluations (Kuhlen 2004).

Informational Self-Defense

The informational self-defense arises from the digital disobedience or constitutes an independent action in the heat of the moment, and serves the preservation of the informational autonomy and the (self-constructed) digital identity (Bendel 2016). For example, you could tear off the data glasses of people walking towards you, because they might record you, could stop cars whose cameras have recorded you and ask for data deletion, or you are as a fake on such platforms that use the personal data for economic purposes. Whether mitigating circumstances or even claims for impunity are to be asserted in the event of damage or infringement will be decided in individual cases. A term with an additional meaning is "digital self-defense".

People will take a stand against face recognition systems. They will cover up themselves, if still legally authorized, they will apply makeup, will get tattooed and affix jewelry, will have optical operations performed and use technical means to try to disrupt and influence the systems. If they do not commit themselves to self-defense, then perhaps to the somewhat weaker concept of information thrift.

The Renaissance of Physiognomy

It becomes obvious that the physiognomy of ancient times, the Middle Ages and the Renaissance has resurrected and finds its representatives and propagators. Above all, the questionable excesses of the Enlightenment and the nineteenth and twentieth century have resurfaced, in which face, race, intelligence and worth were combined.

This development seems quite strange today. In Europe, they rub their eyes when seeing the ghosts that they seem to have successfully banished. In the United States, where diversity plays a major role, where discrimination on grounds of origin, age and gender is ostracized and punished, they see themselves in a great dilemma that is also expressed in the caution of the researchers from Stanford University. Here, social-political claims, whether they are exaggerated or not, clash with technical possibilities. At the same time, in some circles in the US, some states and sensitivities that have arisen in Europe in the course of history may meet with a certain lack of understanding. In spite of this, it could be of interest to them – as well as to researchers from other parts of the world – to study the European idea and intellectual history under these considerations.

What obviously drives this development are economic and political interests. In times of the greatest uncertainty, one hopes more than ever to have simple procedures with which – if it is not simply a question of maximizing profits – the supposed evil can be fought against. This is combined with the potency expected from AI, and with the effectiveness and efficiency of machine processes. In addition to the self-assumed possibilities, opportunities play a role that one can claim in front of others: one can persuade the population that it is possible to fight terror with technical means. Information ethics can use the discursive method to disclose the interests of the parties involved and to help assess the adequacy of the means on all sides (Kuhlen 2004).

Summary and Outlook

Face recognition has become a big topic. Now, its direction is changing more and more. To a large extent, the machine-based approaches in their categorizations and functionalities are very questionable. Thus, moral and legal approaches are messed up, in some places it is suggested that criminals are basically bad people, even though they only violate certain laws. Moreover, it is suggested that the machine can read faces better and faster.

In certain questions such as the sexual orientation, a software seems to actually perform this determination better than a human does. However, as it turned out, the person does not necessarily have an interest in this determination. Moreover, it is also helpful or even essential for the software if it receives additional data that have nothing to do with the face and the head. These, in turn, may be of discriminatory character.

In the end, there are many reasons not to use face recognition at all to determine character traits, personality traits and temperament as well as sexual orientation. At the very least, however, there are many ethical questions that were dealt with in this article to some extent, and which may reverberate in political considerations.

References

Anderson, M.; and Anderson, S. L. eds. 2011. *Machine Ethics*. Cambridge: Cambridge University Press.

Belting, H. 2013. *Faces*. Eine Geschichte des Gesichts. München: C. H. Beck.

Bendel, O. 2017a. Gesichtserkennung. *Gabler Wirtschaftslexikon*. Wiesbaden: Springer Gabler. http://wirtschaftslexikon.gabler.de/Definition/gesichtserkennungssoftware.html.

Bendel, O. 2017b. Neue Spione in den Straßen, auf den Plätzen und in den Läden: Interaktive Werbeflächen aus ethischer Sicht. *Telepolis*, August 15, 2017. https://www.heise.de/tp/features/Neue-Spione-in-den-Strassen-auf-den-Plaetzen-und-in-den-Laeden-3797118.html.

Bendel, O. 2016. *300 Keywords Informationsethik: Grundwissen aus Computer-, Netz- und Neue-Medien-Ethik sowie Maschinenethik.* Wiesbaden: Springer Gabler.

Brien, J. 2016. Gefährliches Spiel: Eine KI hat gelernt, Kriminelle anhand von Fotos zu erkennen. *t3n*, November 24, 2016. http://t3n.de/news/ki-kriminelle-fotos-erkennen-769867/.

Campe, R.; and Schneider, M. eds. 1996. *Geschichten der Physiognomik. Text – Bild – Wissen*. Freiburg im Breisgau: Rombach.

Feng, R.; and Prabhakaran, B. 2016. On the “Face of Things”. *ICMR'16*, June 06–09, 2016, New York, USA.

Kosinski, M.; and Wang, Y. 2017. Deep neural networks are more accurate than humans at detecting sexual orientation from facial images. *Journal of Personality and Social Psychology*, 2017. Preprint via https://psyarxiv.com/hv28a/.

Kuhlen, R. 2004. *Informationsethik: Umgang mit Wissen und Informationen in elektronischen Räumen.* UVK/UTB, Konstanz 2004.

Li, S. Z.; and Jain, A. K. eds. 2011. *Handbook of Face Recognition*. London: Springer.

Marlow, J.; and Wiese, J. 2017. Surveying Surveying User Reactions to Recommendations Based on Inferences Made by Face Detection Technology. *RecSys'17*, August 27–31, 2017, Como, Italy. pp. 269 – 273.

Meyer, J.-B. 2017. So, wie Sie aussehen, sind Sie ein Terrorist! *Computerwoche*, June 7, 2017. https://www.computerwoche.de/a/so-wie-sie-aussehen-sind-sie-ein-terrorist,3229425.

Michel, C. 2016. Rassismus? Beim KI-Schönheitswettbewerb gewinnen fast nur Weiße. *Wired*, September 9, 2016. https://www.wired.de/collection/life/rassismus-beim-ki-schoenheitswettbewerb-gewinnen-fast-nur-weisse.

Morchner, T. 2010. Streit um Kennzeichnung von Überwachungskameras in Hannover. *Hannoversche Allgemeine*, November 23, 2010. http://www.haz.de/Hannover/Aus-der-Stadt/Uebersicht/Streit-um-Kennzeichnung-von-Ueberwachungskameras-in-Hannover.

O'Neil, C. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. New York: Crown.

Schmölders, C. 2007. *Das Vorurteil im Leibe. Eine Einführung in die Physiognomik.* 3th edition. Berlin: Akademie-Verlag.

Schneemann, D. 2002. *Wer bin ich? Wer bist Du?: Das große Buch der Menschenkenntnis*. Bonn-Oberkassel: Heel Verlag.

Schwertfeger, B. 2006. Verräterische Beule am Kopf. *Spiegel Online*, November 6, 2006. http://www.spiegel.de/lebenundlernen/job/personalauswahl-per-gesichtsanalyse-verraeterische-beule-am-kopf-a-446426.html.

Taschwer, K. 2017. Software liest aus Porträtfotos sexuelle Orientierung ab. *derStandard.at*, September 8, 2017. http://derstandard.at/2000063816118/Software-liest-aus-Portraetfotos-sexuelle-Orientierung-ab.

Thomas, I. C. 2017. Dieses Computerprogramm verrät, wie schön Sie sind. *Welt*, October 7, 2017. https://www.welt.de/wirtschaft/webwelt/article150754121/Dieses-Computerprogramm-verraet-wie-schoen-Sie-sind.html.

Wu, X.; and Zhang, X. 2016. Automated Inference on Criminality Using Face Images. *arXiv*, November 13, 2016. https://arxiv.org/abs/1611.04135v1.

Wu, X.; and Zhang, X. Responses to Critiques on Machine Learning of Criminality Perceptions. *arXiv*, May 26, 2017. https://arxiv.org/abs/1611.04135v3.

Maintaining the Humanity of Our Models

Umang Bhatt
Carnegie Mellon University
Pittsburgh, PA 15213, USA
umang@cmu.edu

Abstract

Artificial intelligence (AI) and machine learning (ML) have been major research interests in computer science for the better part of the last few decades. However, all too recently, both AI and ML have rapidly grown to be media frenzies, pressuring companies and researchers to claim they use these technologies. As ML continues to percolate into the layman's life, we, as computer scientists and machine learning researchers, are responsible for ensuring we clearly convey the extent of our work and the humanity of our models. In our current discussion, we limit ourselves to the following three important aspects that are needed to regularize ML for mass adoption: a standard for model interpretability, a consideration for human bias in data, and an understanding of a model's societal effects.

Introduction

Mainstream media, any non-academic or non-research outlet, fawn over the tandem of machine learning (ML) and artificial intelligence (AI). Recently, technologies like AlphaGo, competitions like the Netflix Prize, and once sci-fi fantasies like self-driving cars have dominated news headlines. The media is correct in claiming that, while ML is outperforming humans at clerical and pattern-driven work, the next wave of AI will revolutionize medicine, law, finance, and transportation by processing data more efficiently than humans (Grace and Salvatier 2017). It is not wrong to be proud of and eager about the advances made in these fields annually. AI can be compared to the steam engine and electricity: powerful general-purpose technologies that can forever alter the fabric of society (Brynjolfsson and McAffee 2014). However, it is erroneous to overstate these technologies' capabilities in the immediate future, which we define hereafter as ~1-2 years. AI growth is slowly yet drastically automating aspects of the monotony in our lives (Schwab 2016).

As AI enters the limelight and displaces all, regardless of the color of their collar, researchers and practitioners of the field must poise the resultant models to be interpretable, unbiased, impactful, and thus humane (Kaplan 2015). In our discussion, we define humanity and humane to be the ethereal and emotional impact of these models on humans. We define AI as encompassing its subfield of ML. In order to build a ML system that values humanity, we consider the following questions: (1) How can researchers make their work interpretable for the end user? (2) How can researchers ensure their algorithms are not learning now unlawful or immoral patterns from antiquated data? (3) How can researchers evaluate the societal effects of their predictions?

We believe these three questions provide the foundation needed to succeed in maintaining the humanity of the models we create. To scale to the masses, ML systems need be interpretable to a non-expert. Laymen should be able to understand the sequence of steps and data points used (and their respective weights) to achieve the final result. ML systems must draw from data that researchers have vetted for potential social bias, thus ensuring the fairness of the eventual conclusion. This is an overlooked portion of current ML work: most researchers claim themselves to be data-agnostic; however, it is imperative they care about the features, source, and context of datasets (O'Neil 2016). Finally, ML systems must be aware of the user impact of each prediction made and each pattern found. Having a pointed, narrow goal with low impact is the current rule of thumb to ensure little disruption in other parts of a user's life (Armstrong and Levinstein 2017). To that end, we dive into the need for all three pillars, as the fields of AI and ML continue to evolve.

Model Interpretability

Imagine a patient visiting a doctor in 2030. They walk into an empty room filled with sensors and large screen with necessary instructions. Once the minimum readings have been made (non-invasively and implicitly), the patient can see a diagnosis (e.g. Diabetes) generated automatically by a black box. If researchers are not cognizant of the implications of their predictions, delivering a potentially life-changing diagnosis in such an insensitive manner can stifle the adoption of AI systems, since the system lacks humanity in diagnosis. As Manuela Veloso once said, “If we don’t worry about the explanation [of the result], we won’t be able to trust the systems.” We, as researchers and practitioners, need to ensure our current black box models gain *clear-box* access to allow end users to reason about our prediction. Therefore, researchers must prioritize exposing the inner workings of ML systems to promote interpretability - the explanation behind predictions – thus bringing the world more personable, humane models.

Current State

ML today begets a robust strength in prediction power in decision-making processes (at least in the supervised case, which we assume from here). However, due to a mismatch between prediction objectives (i.e. test set performance) and the real world costs of deployment, there is an unfulfilled demand for interpretability (Lipton 2017). As the final users of ML systems are typically non-experts, models lacking interpretability are rendered ineffective and useless. Though there exists no concrete definition of interpretability, it broadly refers to explaining a model in humanly understandable terms: many desiderata for modern ML systems, like robustness, fairness, and trust, are also commonly grouped with interpretability (Doshi-Velez and Kim 2017).

There exists a need for rigorously standardizing interpretability, since the European Union will prevent automated individual decision-making this year (Goodman and Flaxman 2016). As of now, dimensionality reduction techniques like backward feature selection on a single layer perceptron or feature extraction via principle component analysis suffices to make a model interpretable in simple cases (Vellido 2012). Sparse linear classifiers and discretization methods (decision trees, rule sets, etc.) are well-known interpretable models (Kim 2015). However, much interest now lies in the nonlinear, high dimensional models and related deep learning techniques. Researchers working on joint model training techniques are exploiting known interpretable models to provide laymen with explanations for a given prediction.

More recent techniques have actually implicitly prioritized interpretability, albeit void of a standardization. Researchers working on neural modulation for semantic search in visual content are inherently making some ML models more interpretable by employing explicit reasoning and attention.

Case Study: Medicine

Returning to the 2030 scenario, the patient demands an explanation of how a complex model, like Doctor AI (“a generic predictive model that covers observed medical conditions and medication uses”), came to its diagnosis (Choi et al. 2016). Though the model might be confident about its prediction, it must expose the sequence of decisions that led to the conclusion. One option would be jointly training a recurrent neural network, a long short-term memory (LSTM) per se, with a hidden Markov model (HMM) to expose the HMM state sequences to the end user (Krakovna and Doshi-Velez 2016). This technique leverages both the predictive power of an LSTM and the explicit states of an HMM: this even unlocks transfer learning as an LSTM model trained on a sufficiently large electronic health record can be transferred to any hospital (Choi et al. 2016). However, a major shortcoming of this approach is that a domain expert must be leveraged to name the states of the HMM: it is nearly impossible for a computer scientist to attempt to name a given state sequence of symptoms and vital signs as potentially contributing to a particular diagnosis. In some simpler planning tasks, expert knowledge is taken into account in the prior distribution over the area of interest, but this does not generalize well to all situations (Kim 2015). Nonetheless, coupling combined model training with test set performance on the top-k ICD-9 codes[1] can produce accurate and interpretable results (Lipton and Kale 2015, Nigam 2016). Another such technique for making these predictively powerful LSTMs more explainable is employing input gradients to generalize decision logic, which is irrespective of the dataset (Ross, Hughes, Doshi-Velez 2017). These techniques are all means towards the end of making our ML models more interpretable and thus more humane.

Human Bias in Data

The source and features of data used as a basis for our models are essential to understanding the inherent human bias in a model’s predictions. When productionalizing a model, we must divulge the exact source and features of the data used to train that model. Data, contrary to layman’s thoughts, ages and grows stale. Imagine if researchers used data from the Jim Crow days to predict in which zip codes are people most likely to go to jail again (O’Neil 2016). Overtime, the data from yesteryear becomes irrelevant. So, can researchers not just create a threshold or add a layer of logistic weight to our data by recency? Well, a recency bias is just as unproductive (Abah 2016). Acknowledging the existence of and taking steps to correct

[1] The authors pick the top k most frequent ICD-9 (alphanumeric codes for patient diagnosis) and classify the accuracy of our model on those codes.

this potentially unfair data yields more humane models, as an unbiased model fed biased data gives a biased result.

Current State

When assessing the quality/recency of and reducing the human bias of a dataset, two techniques are common. One technique is debiasing, which manually severs the learned relationship between two entities. In example, gender bias in natural language generation from processing/training on text corpuses is all too common. A gender bias-free dataset of images can be created when we place constraints on certain relationships between entities within the images (Zhao et al. 2017). In a text generation algorithm, gender bias can be mitigated by identifying known gender biased words, working in a gender neutral subspace, and understanding the distance of a gender neutral world towards the preidentified gender subspaces (Bolukbasi et al. 2016). Another technique is simply omission of the stale or biased data from training; it is trivial to state, but such a decision is lossy and certain patterns in the data will be missed.

It is crucial to note that in both scenarios, researchers are imposing their own bias and morality on a given problem space. For example, if researchers think (or even empirically show) that zip code of residence is a high predictor of where crime occurs, they are then faced with a moral struggle of whether or not to patrol more in those zip codes, disadvantaging the portion of non-criminals in a zip code deemed crime prone. The legality of models matters considerably as an ounce of human bias can violate the law (Samek 2017). To that end, we show a need to remove human bias disparities with as little impact on accuracy as possible (Johndrow and Lum 2017).

Case Study: Recidivism

Recidivism prediction (that is, the propensity of a person to return to jail once released) is bursting with social bias. Though models like PredPol[2] exist, there is no formal feedback loop for all involved parties; thus, there exists a lack of randomness in the data (Ensign et al. 2017). Without this randomness, a human bias is propagated in the data (e.g. only patrol neighborhoods of criminals who are currently imprisoned). Unfortunately, researchers lack a method to understand the fairness of their predictions, other than the false positive rates of two subgroups within the population in question. One suggestion is to optimize parameter instability and disparity (Chouldechova and G'Sell 2017). More interpretably, one can perform a subset scan to detect if a given class has noteworthy bias for in a given subgroup (Zhang and Neill 2017). Such techniques only arise if researchers heed human biases in data, which will be of utmost importance as ML adoption continues to skyrocket.

Societal Effects

The output of ML systems affects real flesh and blood beings. Unfortunately, all too often, researchers lose sight of this reality. Some researchers focus on optimizing objectives on benchmark datasets instead of the real world applications of the code they write (Wagstaff 2012). They want to be able to transfer their expertise and models to new domains, wherein ML can augment archaic practices and automate pattern-based predictions. For example, clothing companies no longer use only intuition and actuarial science to forecast their products' performance, instead they also use models that incorporate seasonality, user preferences, and industry trends to decide what type of clothing should be designed next season (Brynjolfsson and McAffee 2014). In confluence with the proliferation of ML use cases, we must remain cognizant of the legality of our models and predictions and be alert of user intent and reception.

Current State

Society benefits from ML models daily. These models tell us what stocks to buy, how much demand a restaurant can expect next quarter, what country poses the most threat to another, whom we should date, etc. (Ross 2016). Society seems like it is subject to the output of these models, and thus mainstream media often misinterprets the power of ML.

For example, in the realm of natural language processing, many recent works report that in multi-agent environments, where agents communicate via strings of tokens to perform a given task, grounded and compositional language naturally emerges. Though this may be the case in controlled circumstances, we cannot generalize this to say: "AI agents make their languages and thus we need to shut them down," as many media claim (Lewis et al. 2017). Upon review, it becomes evident that language cannot emerge naturally and systems are shut down due to a lack of human interpretability: that is, one AI agent may say "Red man ball sit!" to another agent, who understands that to mean "Hello, how are you?" in English – without human intervention, the agents communicate in a nonsensical, incomprehensible grammar, basically gibberish, thus stressing the need for the first pillar of interpretability (Kottur et al. 2017).

As mass ML adoption is imminent, being mindful of such misinterpretations and effectively communicating the limits of ML must be kept at the top of our minds.

[2] PredPol allows law enforcement to predict where crime will happen given historical/real-time data feeds and then assigns patrol units accordingly (Ensign et al. 2017).

Case Study: Pricing

In the e-commerce world, companies optimize models to maximize profit or increase purchase frequency. One such model is a dynamic pricing engine, which prices goods based on the targeted consumer's willingness to pay. As such, these engines are used to serve the *optimal* price for a given user to maximize company profits. Plagued by sparse user level data and by legal constraints on what features can and cannot be used, dynamic pricing experts manage programs like time-limited coupons forecasted via a point-process model that makes real-time, global estimates based on transaction history and patterns (Manzoor and Akoglu 2017). Such pricing programs must be interpretable and unbiased; if they are not, the societal consequences of erroneous prices (or worse, of price discrimination) are catastrophic for a company. Being aware of and responsive to the implications of ML models is the final key towards more humane and adoptable models.

Conclusion

To be prepared for mass adoption of machine learning systems, we, as researchers and practitioners, must adopt a framework for developing humane models that ensure interpretability, unbiasedness, and practicality. By creating a rigorous standard for machine learning interpretability, we can transform the medical predictive analytics industry. By understanding the inherent human bias in the data we collect and the sample it represents, we can ensure that we build a more unbiased model for police patrol. By thinking deeply about the societal effects and ethicality of our predictions, we can ensure we deliver profitable and fair prices in the e-commerce industry. All three pillars can displace society's perception of machine learning, as the true power and beauty of how we can use autonomous agents and machine learning comes to fruition when we maintain the humanity of our models.

Acknowledgements

Many thanks to my advisors/mentors at CMU: Jose M.F. Moura, J. Zico Kolter, and David O'Hallaron for their valuable discussions and to my research partners: Satwik Kottur, Edgar Xi, Shouvik Mani, and Sam Fazel for their continued support.

References

Abah, J. 2016. Recency Bias in the Era of Big Data: The Need to Strengthen the Status of History of Mathematics In Nigerian Schools. *In Advances in Multidisciplinary and Scientific Research Journal*.

Armstrong, S., and Levinstein, B. 2017. Low Impact Artificial Intelligences. *arXiv: 1705.10720*

Bird, S., Barocas, S., Crawford, K., Diaz, F., and Wallach, H.. 2016. Exploring or Exploiting? Social and Ethical Implications of Autonomous Experimentation in AI. *Workshop on Fairness, Accountability, and Transparency in Machine Learning, 2016*. New York, NY.

Bolukbasi, T., Chang, K., Zou, J., Saligrama, V., Kalai, A. 2016 Man is to Computer Programmer as Woman is to Homemaker? Debiasing Word Embeddings. *arXiv: 1607.06520*

Brynjolfsson, E., and McAffe, A. 2014. The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies. *WW Norton & Company.*

Choi, E., Bahadori, M.T., Schuetz, A., Stewart, W.F., Sun, J. 2016. Doctor AI: Predicting Clinical Events via Recurrent Neural Networks. *In Proceedings for 2016 Machine Learning and Healthcare Conference.* Los Angeles, CA

Chouldechova, A. and G'Sell, M. 2017. Fairer and more accurate, but for whom? *In Proceedings for FAT/ML 2017.* Halifax, NS, Canada.

Doshi-Velez, F., and Kim, B. 2017. Towards A Rigorous Science of Interpretable Machine Learning. *arXiv: 1702.08608*

Doshi-Velez, F. Kortz, M, et. al. Accountability of AI Under the Law: The Role of Explanation. *arXiv: 1711.01134*

Ensign, D., Friedler, S., Neville, S., Scheidegger, C., Venkatasubramanian, S. 2017 Runaway Feedback Loops in Predictive Policing. *In Proceedings for FAT/ML 2017*. Halifax, NS, Canada.

Frank, B. September 19, 2017. You might use AI, but that doesn't mean you're an AI company. *VentureBeat*.

Goodman, B. and Flaxman, S. 2016. European Union regulations on algorithmic decision-making and a "right to explanation". *In Proceedings for 2016 ICML Workshop on Human Interpretability in Machine Learning,* New York, NY.

Grace, K., Salvatier, J., Dafoe, A., Zhang, B., Evans, O. 2017. When Will AI Exceed Human Performance? Evidence from AI Experts. *arXiv: 1705.08807*

Grbovic, M., Radosavljevic, et. al. 2016. E-commerce in Your Inbox: Product Recommendations at Scale. *In Proceedings for KDD 2015*. Sydney, Australia.

Johndrow, J. and Lum, K. 2017. An algorithm for removing sensitive information: application to race-independent recidivism prediction. *arXiv: 1703.04957*

Kaplan, J. 2015. Humans Need Not Apply: A Guide to Wealth and Work in the Age of Artificial Intelligence. *Yale University Press.*

Karpathy, A. May, 31, 2017. AlphaGo, in context. *Medium.*

Kim, B. 2015. Interactive and interpretable machine learning models for human machine collaboration. PhD diss., Massachusetts Institute of Technology, 2015.

Kottur, S., Moura, J., Lee, S., Batra, D. 2017. Natural Language Does Not Emerge 'Naturally' in Multi-Agent Dialog. *In Proceedings for EMNLP 2017.* Denmark.

Krakovna, V. and Doshi-Velez, F. 2016. Increasing the Interpretability of Recurrent Neural Networks Using Hidden Markov Models. *In Proceedings for NIPS 2016 Workshop on Interpretable Machine Learning in Complex Systems*. Barcelona, Spain.

Lewis, M., Yarats, D., Dauphin, Y., Parikh, D., Batra, D. 2017. Deal or No Deal? End-to-End Learning for Negotiation Dialogues. *arXiv: 1706.05125*

Lipton, Z. 2017. The Mythos of Interpretability. *In Proceedings for 2016 ICML Workshop on Human Interpretability in Machine Learning,* New York, NY.

Lipton, Z., Kale, D., Elkan, C., Wetzel, R. 2015. Learning to Diagnose with LSTM Recurrent Neural Networks. *arXiv: 1511.03677*

Manzoor, E., and Akoglu, L. 2017. RUSH! Targeted Time-limited Coupons via Purchase Forecasts. *In Proceedings for KDD 2017*. Halifax, NS, Canada.

Nigam, P. 2016. Applying Deep Learning to ICD-9 Multi-label Classification from Medical Records. *Stanford University*

O'Neil, C. 2016. Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy. *Broadway Books.*

Reese, H. 2016. Transparent machine learning: How to create 'clear-box' AI. *Tech Republic.*

Ribeiro, M., Singh, S., Guestrin, C. 2016. Nothing Else Matters: Model-Agnostic Explanations By Identifying Prediction Invariance. *arXiv:1611.05817*

Ross, A. 2016. The Industries of the Future. *Simon & Schuster Paperbacks.*

Ross, A., Hughes, M., Doshi-Velez, F. 2017. Right for the Right Reasons: Training Differentiable Models by Constraining their Explanations. *arXiv:1703.03717*

Schwab, K. 2016. The Fourth Industrial Revolution. *Crown Business.*

Samek, W., Wiegand, T., Muller, K.R. 2017. Explainable Artificial Intelligence: Understanding, Visualizing and Interpreting Deep Learning Models. *arXiv: 1708.08296*

Vellido, A., Martin-Guerreo, J., Lisboa, P. 2012. Making Machine Learning Models Interpretable. *In Proceedings for European Symposium on Artificial Neural Networks, Computational Intelligence, and Machine Learning 2012*. Bruges, Belgium.

Wagstaff, K. 2012. Machine Learning that Matters. *In Proceedings for the 29th International Conference on Machine Learning.* Edinburgh, Scotland, UK

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., Chang, K. 2017. Men Also Like Shopping: Reducing Gender Bias Amplification using Corpus-level Constraints. *arXiv: 1707.09457*

Zhang, Z. and Neill, D. 2017. Identifying Significant Predictive Bias in Classifiers. *arXiv: 1611.08292*

The Heart of the Matter: Patient Autonomy as a Model for the Wellbeing of Technology Users

Emanuelle Burton
Dept of Computer Science
University of Illinois at Chicago

Kristel Clayville
Philosophy and Religion Dept
Eureka College

Judy Goldsmith
Dept of Computer Science
University of Kentucky

Nicholas Mattei
TJ Watson Research Center
IBM

Abstract

We draw on concepts in medical ethics to consider how computer science, and AI in particular, can develop critical tools for thinking concretely about technology's impact on the wellbeing of the people who use it. We focus on patient autonomy—the ability to set the terms of ones encounter with medicine—and on the mediating concepts of informed consent and decisional capacity, which enable doctors to honor patients' autonomy in messy and non-ideal circumstances. This comparative study is organized around a fictional case study of a heart patient with cardiac implants. Using this case study, we identify points of overlap and of difference between medical ethics and technology ethics, and leverage a discussion of that intertwined scenario to offer initial practical suggestions about how we can adapt the concepts of decisional capacity and informed consent to the discussion of technology design.

Introduction

Machines will be making life-and-death decisions for individuals in the near future, as well as decisions that have a profound impact on the quality of human lives. Not only will they drive vehicles and deliver aid, they may triage disaster victim rescues and hospital admissions, they will control thermostats, schedule emergency services, help farmers predict weather and timing of growing seasons, work with food processing plants' supply chains, adjudicate insurance and parole claims, and decide who has access to emergency shelters in the wake of natural disasters. In ways both large and small, current and in-development applications of AI are altering the basic conditions of ordinary human experience, from the imminent availability of self-driving cars to robot companions for the elderly (Sabanovic et al. 2013) or the robophilic (Danaher and McArthur 2017).

All of these AI-driven decisions are necessarily predicated on comparative value judgments about human worth and human goods: the importance of children's lives vs. seniors' lives in a natural disaster, or the value of students' security vs. their personal dignity at a high-risk high school, or the appropriate course of medical care for a terminally ill patient who is physically and emotionally suffering. These are the same value judgments that transplant teams make every time they prepare to operate. Whether those values are predetermined by developers or companies or are learned by example through machine learning algorithms, these mechanized decisions—and the substrate of comparative values that structure the automated decision-makers—will have a profound impact on people's lives and wellbeing.

But what exactly makes a human life valuable and distinctive? What qualities of internal self or external environment need to be in place for a person to be able to live and act as a person? How do particular changes to their environment enhance, or circumscribe, their ability to be a version of themselves that they recognize and prefer? For most technologists who understand their work as a way to improve human lives, the importance of those lives and the reasons why they matter have been largely a product of moral intuition rather than of carefully-defined principles. Such intuitions are difficult to formalize in a way that can be programed directly or entrusted to an algorithm to learn by example, particularly in the absence of a conceptual language that can identify them or draw distinctions between them.

The proliferation of AI in daily life makes it ever more vital and pressing that technologists can think specifically about those aspects of the person that make them recognizable and distinct as people, and furthermore how those human qualities are amenable to improvement, or vulnerable to harm, through specific changes in the conditions of daily life (Burton, Goldsmith, and Mattei 2015; Burton et al. 2017). Furthermore, *it is imperative that AI ethics develop its own conceptual tools that can account for the particular ways in which AI can impact those conditions of daily life.* So equipped, technologists will be able to discuss the parameters and significance of the interventions that their designs are making, and to think more concretely about how design and programming choices can protect and enhance the lives of individuals and societies.

This aim—to enhance, rather than diminish, human lives through technology—is made particularly difficult by the knowledge gap between those who build and maintain the technologies and those who use them without the technological expertise to understand how they work. Such nonspecialists users face several disadvantages, even with respect to technologies and platforms that are designed for non-specialist use such as a smartphone or Twitter. Not only are these users less likely to be aware of potential security breaches, the signs of such breaches, or the steps they might take to prevent them; they are also far less likely to be aware of any modifications that would enable these users to fine-tune their experience for their own personal comfort and convenience. Thus, *even at the level of everyday personal technology use, there exists a significant power imbalance between technology experts and non-experts*. The depth and scope of that power imbalance grows exponentially if one also considers those experts' professional work designing, building and maintaining the systems that other users rely on but lack the expertise to understand.

This expertise-based power imbalance, while particularly pressing in technology ethics (and perhaps AI in particular), is not unique. A similar power imbalance has long existed in medicine, a field whose practitioners need extensive specialist knowledge even as they serve a user base of patients who mostly lack that knowledge. Because of the power imbalance implicit in the vast majority of patient-practitioner relationships, patients are often prevented from making choices about their own care even when doctors or nurses are at pains to leave the choice in the patient's hands (Henderson 2003). *To mitigate this problem, medical ethics has developed a family of concepts and practices to help its expert practitioners to navigate the inevitable imbalance in power and knowledge* (Quill and Brody 1996). As this paper will demonstrate, these concepts can be usefully imported (with some significant revision) into technology ethics (Johnson 2009), and can be used to identify specific technology design practices that preserve non-expert technology users' capacity for self-determination.

Contribution. In this paper, we described the concept of patient autonomy from medical ethics, as well as the corollary concepts of informed consent and decisional capacity. We use a fictional case study to highlight the both the points of intersection and points of divergence between traditional medical ethics concerns and technology ethics concerns. We then, on the basis of case study discussion, develop working definitions of informed consent and decisional capacity that are attuned to the central problems facing technology ethics. Finally, on the strength of these newly-adapted concepts, we will offer some concrete examples of how current projects in AI and technology are working to support human autonomy, or how they could be adapted to better support it.

Autonomy in Medical Ethics

Most western medical practitioners would identify **autonomy** as the central tenet of medical ethics. Autonomy is the principle that mandates **respect for persons**, meaning that individuals have free exercise with regard to whether and what kind of treatment to receive, and honoring this independence is central to contemporary medical ethics (Jonsen, Siegler, and Winslade 2015) . Patient autonomy as a governing concept in medical ethics is relatively recent; the shift toward it and away from medical paternalism was fueled both by broader social movements that sought to empower the individual and by the development of a more consumerist model of medicine as physicians sought to protect themselves from malpractice (Billings and Krakauer 2011).

In practical medical ethics, the term autonomy has two distinct uses, which are related but which also operate independently of each other. The first usage is to affirm that the patient deserves autonomy, the power to exert influence over what happens to them; the second usage concerns the question of whether the patient is able to exercise that autonomy. Because people frequently seek medical care at a moment when they are mentally and physically compromised, it is not enough to affirm that a patient deserves autonomy. It is necessary for medical providers to take deliberate steps in order to protect the patient's autonomy, and ensure that the patient is able and empowered to make decisions that reflect their wishes.

Neither dimension of autonomy—autonomy-as-recognition or autonomy-as-exercise—simply exists as a given. Because of the systemic power imbalance between expert care providers and their nonexpert patients, two important constraints have been put in place to ensure that the patient's autonomy is honored not only in principle but in practice. They are **informed consent** and **decisional capacity**. In the United states, when a patient undergoes a medical procedure, that patient must consent to it, and that consent must follow a conversation in which the doctor explains the procedure's risks, benefits to the patient, as well as other treatment options. After this conversation has happened, the patient signs a document acknowledging that this conversation took place, and the patient is thereby giving *informed* consent to the procedure. Because informed consent documents a conversation, *it is approached as a process rather than a one-time event*. Patients can change their minds at any point leading up to or during the procedure.

No medical procedures or treatments should be undertaken without informed consent, but only patients who have decisional capacity can give informed consent. In general, adult patients are presumed to have decisional capacity, but there are categories of patients who lack it. Patients can lack decisional capacity due to age (children), medical status (dementia patients), temporary states (sedated), or institutional status (prisoners). But this absence of decisional capacity is not permanent; children will age into being decisional and able to give informed consent, sedated patients will wake up, and prisoners may be freed, thus enabling them to make decisions free of coercion.

Paradoxically—or so it seems at first—these limits on a patient's decision-making were instituted precisely to preserve the patient's autonomy, because they place limits on a doctor's ability to manipulate patients into undergoing treatments. The constraints were developed in response to abuses of paternalism, and were designed to constrict doctors' freedom by preventing them from taking advantage of patients who were, for whatever reason, unable to exercise their own autonomy.

As medical culture has evolved toward being more patient-centered, the language and conceptual framework of autonomy have likewise been enhanced to focus more on how patients can exercise autonomy, rather than on the constriction of the doctor's. Patients can, in fact, prepare for a future in which they are non-decisional, by creating legal documents that spell out their wishes, should they be incapacitated. They can also cede decision-making power to specified others, for such an eventuality. In the absence of such explicit and legally binding instructions, it is assumed in most societies that a surrogate decision maker from the family can speak for the patient's wishes.

As we will argue, the concept of patient autonomy—and its concepts of informed consent and decisional capacity—offer a useful model for technology ethics in thinking about how to preserve and enhance the wellbeing of technology users. As the above discussion illustrates, however, the core problems in medicine are not identical to those in technology. *In order for these imported concepts of autonomy, decisional capacity, and informed consent to be useful to technology ethics, they need to be adapted, but in a way that preserves the element that makes them useful.* We use the following fictional case study to illustrate points of overlap and divergence.

Case Study

Consider a heart patient, Joe, who has two implants to help with his heart: a pacemaker, which regulates his heartbeat, and an implantable cardioverter defibrillator (ICD), which can restart his heart if it stops. This is a common case in the US with over 947 heart related implants per million people (Mond and Proclemer 2011). Some years ago, in consultation with his doctor (as is legally required), Joe requested and was granted Do Not Resuscitate (DNR) status. At a recent doctor's visit, Joe was told that restarting his heart would be intensely painful, and that in such an event, his heart would likely fail and need to be restarted repeatedly. Given his DNR status, Joe's doctor asked whether Joe wants the ICD turned off.

Joe's case raises a set of questions that are common to many medical ethics case studies, most of which center around autonomy.

1. Does Joe have the right to make these decisions? If he is in pain, can his judgment be trusted?
2. Do Joe's previous decisions express a state of mind that is still binding for the present?
3. For Joe's doctor, is there a meaningful difference between Joe refusing aggressive CPR (an external treatment) and refusing an ICD?
4. For Joe, is there a meaningful difference between refusing an ICD and turning off an ICD that is already implanted?
5. For both Joe and his doctor, would turning off the ICD be comparable to euthanasia?

The framing of these questions presumes the concept of autonomy: that Joe deserves the right to determine what happens to him, and that this right to self-determination must be preserved in balance with medicine's broad imperative to preserve and extend life whenever possible. Joe's right to refuse treatment is recognized, but so is the fact that the very conditions of his treatment may mean that he is not decisional, and thus not fit to make decisions that may harm his person.

But as technologists and those thinking about technology ethics will immediately recognize, this slate of questions excludes some important issues, including issues that might be understood in terms of autonomy. Other questions should be raised pertaining to the security of Joe's personal information and self-direction that are directly influenced by the specific technologies that are now part of his body.

1. Who is responsible for implanting and maintaining Joe's machines?
2. What risks are there to Joe in having his cardiac data possibly transmitted by WiFi and stored online?
3. What risks are there in allowing an off-site monitor to control the pacemaker?
4. Should any of the defibrillator itself, a control system, or a human monitor be able to decide to not resuscitate Joe?

Like the medically-oriented questions, these technological questions also recognizably concern Joe's autonomy as a patient/technology user. The underlying premises of the technology ethicist's questions recognize Joe as an entity deserving of the same sort of autonomy accorded to him by the medical ethics list. But there are two key differences between them. The first is that these questions expand the sphere of Joe's autonomy (in the autonomy-as-recognition sense) to include concerns about his personal information and to consider a wider range of possible agents who might impact Joe's wellbeing. The second difference is that, while these questions broaden the scope of Joe's autonomy as something for professionals to worry about, they constrict its actual exercise by the patient himself (in the autonomy-as-exercise sense). In focusing—appropriately and necessarily—on systems-level concerns such as information security and encryption of medical data, *these questions leave little room for Joe's ability to make decisions for himself, or even to understand what is at stake in the decisions he might make.* Although the questions are about the sphere of Joe's autonomy, they do not create or identify an opportunity for him to exercise it.

The contrast between these sets of questions highlights both how medical ethics could refine its notion of autonomy in conversation with technology ethics, and how technology ethics stands to benefit from an imported version of autonomy from medical ethics. With respect to the first dimension of autonomy—recognizing what the patient deserves as a person—technology ethics usefully broadens the sphere of Joe's autonomy insofar as it broadens the scope of things in the world that are not only *his* but *him:* his pacemaker and defibrillator, perhaps even his data. In an age when medicine is increasingly reliant on networked technology and data, medical ethics would do well to learn from technology ethics' reconfiguration of autonomy.

Yet technology ethics is less well equipped than medical ethics to attend to the second aspect of autonomy, the patient's right to determine what happens to him. A concern for Joe's right to exercise his own particular preferences might lead to questions such as the following: Does Joe understand the capabilities and risks (either to his body or his data) of the devices that have been implanted within him, to a degree that he can make an informed decision about them? Is he aware of the experiences of other patients with similar implantations? Does he feel able to ask his doctors to shut off the implanted devices, to opt out after opting in?

We argue that these are the sorts of questions technologists need to be asking, in particular, the designers of AI technologies that can manage the content of a user's online experience or automatically transmit sensitive medical data to doctors. Because technology is necessarily systems-oriented in its approach, the challenges in making room for users' autonomy-as-self-direction are different—and, arguably, even more difficult to overcome—than those in medicine. Therefore, it is not helpful for technology ethics to simply adopt the concept of autonomy from medical ethics unmodified. And yet, if the human wellbeing of technology users—technology ethics' equivalent of patients—is not to fade from view, it is crucial to identify and clarify a notion of autonomy that technologists *can* use, a definition that is analogous to that in medical ethics but more closely keyed to the problems faced in technology ethics. As technology increasingly sets the conditions for human life, not only in medicine but in the public and private sphere, this sort of working definition will prove crucial for technologists who wish to preserve a space for the exercise of autonomy.

Reframing Autonomy for Technology

As our case study indicates, the notion of patient/user autonomy is relevant for technology as well as for medicine, even though the precise contours are different. As human lives are increasingly managed at both macro- and micro-level by smart technologies —and as medical technology itself advances—it becomes pressing for technologists to consider how to enhance (or at least to preserve) users' autonomy. To do so, technologists must consider not only users' right to make decisions for themselves (the first aspect of autonomy), but the conditions that enable them to exercise that autonomy (the second aspect).

In addition, technology ethics also faces some particular hurdles in incorporating user autonomy into existing frameworks of inquiry. As is seen when we compare the two sets of questions in our case study, the very nature of technological work is already an impediment to conceiving of persons in terms that recognize and extend their ability to exercise their autonomy. These hurdles are particularly difficult to overcome in the case of AI, which outsources both large- and small-scale decision-making to programmed learners—and sometimes in ways that are designed to "solve" the idiosyncrasies of users' exercise of their self-directing autonomy (Rapoport 2013).

A further challenge to technology ethics is that there is rarely an appointed human mediator between the user and the technological establishment as there is in medicine. Medical ethics is structured around the relationship between patient and care provider, and this can invest the individual care provider with particular duties and responsibilities. Any useful adaptation of patient/user autonomy needs to assign responsibility in a manner that is both ethically and practicably plausible.

The concept of user autonomy can be rendered more manageable when we approach it by way of of informed consent and decisional capacity. As discussed above, these two concepts were developed in medical ethics as a means to preserve the patient's autonomy when her capacity to exercise that autonomy is in some way compromised. Informed consent and decisional capacity function essentially as "sluice gates" to make sure that the patient/user's autonomy is maintained even in the presence of disruptive or distorting factors.

Informed Consent

In a medical context, *informed consent* helps to preserve the patient/user's autonomy by requiring the doctor to keep the patient apprised of relevant information, and permitting the patient to rescind consent at any point. Informed consent presumes a user who never develops expertise of her own, and is not penalized for it; the burden remains on the expert-provider to communicate clearly and consistently with the user, to ensure she understands and that her wishes are being honored. While this is not the norm in technology we are starting to see ideas like this appear. For example, the Android operating system's reliance on *permissions* for apps which can be granted or revoked from an easy to find screen (Andriotis, Sasse, and Stringhini 2016).

Informed consent presents deep challenges to the basic design principles of technology, because it is deliberately inefficient and resistant to closure. First, it prioritizes certainty that the patient/user understands over the efficient delivery of information. Second, by allowing the patient/user to opt out at any point, it mandates a structure in which processes are begun but never completed, both because patient/users sometimes withdraw consent partway and because even consenting patients/users retain the option to withdraw consent.

But this inefficiency is absolutely vital if the patient/user's autonomy is to be preserved. Because efficiency requires that certain decisions or functions take place en masse for a group of entities without stopping to consult each one, some kinds of efficiency cannot coexist with informed consent. The smarter and more seamless a technology becomes, the more deliberate the technology designer has to be about maintaining space within it for this sort of inefficiency. For example, a massive push update to a high-tech medical implant will be much easier to accomplish if the manufacturers assume that the patient/users have already consented simply by having the device implanted. If, however, a patient's condition or wishes have changed, she might not want her implant to be updated.

It is important to note that not all kinds of efficiency are necessarily at odds with informed consent. Many forms of automation increase the efficiency with which the user's goals are achieved without eclipsing her ability to revise her goals or judgments. There is no need for a given technology to build in opportunities for ongoing consent when that technology executes tasks the users already understand and intend to perform, such as washing dishes or taking depth or temperature measurements.

Whenever technological efficiency is achieved by eliminating the need for the user's input, there is a real risk that the user's autonomy could be compromised. Any technology that makes decisions for its users—even when those decisions are based on prior expressions of consent or preference—is one that has the potential to violate users' autonomy. Although the efficiency of self-monitoring thermostats and smart surveillance technologies is one of their main selling points to users, that very ease of use is what makes it possible for those users' autonomy to be compromised, when their personal data is transmitted in a manner they are not comfortable with or their home monitoring systems do something they dislike. Indeed, for a device or platform to incorporate informed consent in a meaningful way that it must preserve some kinds of inefficiency. The fact that this notion may present a challenge to the normal way technology developers think underscores the need for a concrete concept by which technology ethicists can assert why it is necessary to constrict some kinds of efficiency in order to preserve or enhance users' wellbeing.

In considering what kinds of inefficiency are important for maintaining informed consent, it is helpful to look back to the original concept in medical ethics. In medicine, the deliberate inefficiency of informed consent affords the patient the time to consider (and reconsider) her options in terms of her values and goals. It also forces the care provider to support the patient in this process, rather than imposing decisions upon her. Because the patient's goals or preferences might shift over time or due to changes in her circumstances, the efficient option—taking the patient's initial goals and decisions as a presumptive guide to the future—would undermine her autonomy. Such changes in goals or preferences can be understood as "human" inefficiencies: inefficient or unpredictable movements of character or goals that are essential to a person's autonomy and crucial to preserving their wellbeing. In a medical context, informed consent protects the patient's autonomy by preserving ongoing ability to express her preferences, even when it renders her overall program of care more inefficient. The efficiency of the treatment process is valuable as long as it preserves or enhances the autonomy of the patient/user, and is potentially damaging to her autonomy insofar as it imposes efficiency on the messy and inefficient processes of self-determination.

Therefore, a usable concept of informed consent for technology ethics is one that enables technologists to consider the specific ways in which a given technology creates efficiency. Does it smooth the user's path to a goal she understands and wants? Does it equip her to understand which sort of determinations are being made for her by automated processes, and to single out the determinations that matter to her for further scrutiny and input? Does it create space for her to revise her engagement with it, should her goals or preferences change? With such questions in mind, a technologist is better prepared to evaluate which kinds of efficiency might categorically interfere with a user's autonomy, which ones require ongoing user input of some kind, and which functions can best serve the user in silent efficiency.

Decisional Capacity

Like informed consent, the notion of *decisional capacity*—the recognition that autonomous users are sometimes not in a state to exercise their own autonomy—can be adapted to technology ethics as a means to preserve and enhance user autonomy. As noted above, medical doctors use a range of criteria to determine whether a patient is decisional, but those criteria have two common denominators: they expect the decisional patient/user to make choices in a manner consistent with their previous character and preferences, and they expect any departures from that prior consistency to be "reasonable"—that is, in line with socially-determined ideas.

Decisional capacity in medical settings is typically binary in nature, because the patient/user's role in the relevant medical process is widely understood to be one of consent, rather than execution. (See, for instance, (Jeste, Palmer, and et al. 2007).) If heart patient Joe decides that he wants his ICD turned off, his decisional capacity depends only on whether he is currently capable of making the decision: a medical expert (either Joe's doctor or an ICD specialist) will implement the decision. Joe will be the one to live with the consequences of his choice—which is why he must be decisional in order to make the choice—but his capacity to execute that decision is not a relevant factor. if Joe's judgment is sufficiently consistent with himself, and/or with what is "reasonable," to make what his doctor deems to be a clear-headed decision, then his decision is medically legitimate.

Technology complicates this notion of decisionality because, in most cases, users are also in charge of implementing their decisions. While technology use is not (usually) as complicated as a surgical procedure, some binding End User Agreements (EULAs) are. Additionally, it can require some deftness of body and mind to manipulate a device with an injured hand, or to craft a rejoinder tweet while in a state of righteous outrage that will not cause regret in an hour. Like medicine, technology is a sphere that can magnify the consequences of a given decision; but unlike medicine, technology empowers users to act *without* the mediation of an expert practitioner who can clarify the scope or the stakes of the user's action.

In most cases, the fact that technology extends the scope of its users' ability to act is the primary virtue. The fact that users are able to take these actions instantly, or near-instantly, is further evidence of the quality of a piece of technology. But these same qualities make users particularly vulnerable to undertaking actions whose technologically-augmented scope exceeds the user's capacity to assess the consequences in the moment of decision. It therefore seems not only helpful but necessary to adapt the notion of decisional capacity for use in technology ethics.

In order to be optimally useful for technology ethics, the notion of decisional capacity needs to be expanded to account for the user's role in implementing their own decisions. It can be helpfully recast for technology ethics as *decisional-executive capacity*, incorporating a second layer that raises the question of whether the user is fit, in a given moment, to undertake an action in a manner that they will be happy with later. Examples of at least checking for this include automatic tone alerts for angry emails and Slack warnings before a message is sent to everyone at the workplace.

Decisional capacity creates an opportunity for AI to enhance the autonomy of technology users and medical patients. As noted above, decisional capacity is quite imperfectly realized in a medical context, as doctors are far more likely to deem a patient decisional if the patient agrees with them. An AI, however, is less likely to succumb to this bias (Hurst 2004). While a doctor's ingrained biases can compromise her assessment of whether her patient is decisional, the doctor-patient relationship is nonetheless a useful model for the AI-user relationship in one key respect. While consistency (the first criterion for determining decisionality) is best judged only with respect to the patient himself, the reasonableness of his wishes (the second criterion) is more broadly culturally determined; what seems like a good reason in one society may seem bizarre in another. Because the human doctor will be influenced by the same broad cultural norms, she is well-positioned to assess whether the patient's expressed wishes fit within those cultural norms, though she is also less likely to be sympathetic to reasons that do not fit those norms. In contrast, an AI that determines decisionality could be structured on universal terms, the ideal approach might call for an AI to learn primarily from local data in order to better assess the reasonableness of expressed wishes.

Conclusion

There is a pervasive societal disease about artificial intelligence that ranges from fears of loss of jobs for humans to terror that we will be displaced entirely by self-aware, higher-functioning AIs. One strain of this anxiety is that the machines will be programmed with more concern for efficiency than for the wellbeing of the humans they are designed to serve. But these things are not determined yet. What is necessary to balance the drive toward efficiency is a focus on how AI can support the distinctively human qualities of its users. We believe that engineers and computer scientists can learn from medical ethicists, and provide a vital viewpoint to the field of medical ethics itself. Through this and broader communication throughout the industries and domains where AI is applied will ensure that AI can live up to the potential envisioned by its boosters, and become a vital part of the architecture of a better human future.

References

Andriotis, P.; Sasse, M. A.; and Stringhini, G. 2016. Permissions snapshots: Assessing users' adaptation to the android runtime permission model. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–6.

Billings, J. A., and Krakauer, E. L. 2011. On patient autonomy and physician responsibility in end-of-life care. *Arch Intern Med* 171(9):849–853.

Burton, E.; Goldsmith, J.; Koenig, S.; Kuipers, B.; Mattei, N.; and Walsh, T. 2017. Ethical considerations in artificial intelligence courses. *AI Magazine* Summer.

Burton, E.; Goldsmith, J.; and Mattei, N. 2015. Teaching AI ethics using science fiction. In *1st International Workshop on AI, Ethics and Society, Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.

Danaher, J., and McArthur, N., eds. 2017. MIT Press.

Henderson, S. 2003. Power imbalance between nurses and patients: a potential inhibitor of partnership in care. *Journal of Clinical Nursing* 12(4):501–508.

Hurst, S. A. 2004. When patients refuse assessment of decision-making capacity: How should clinicians respond? *ARCH Internal Medicine* 164(16):1757–1760.

Jeste, D.; Palmer, B.; and et al., P. A. 2007. A new brief instrument for assessing decisional capacity for clinical research. *Archives of General Psychiatry* 64(8):966–974.

Johnson, D. G. 2009. *Computer Ethics*. Pearson, 4th edition.

Jonsen, A. R.; Siegler, M.; and Winslade, W. J. 2015. *Clinical Ethics: A Practical Approach to Ethical Decisions in Clinical Medicine. 8th ed.* McGraw Hill Education.

Mond, H. G., and Proclemer, A. 2011. The 11th World Survey of Cardiac Pacing and Implantable Cardioverter-Defibrillators: Calendar Year 2009–A World Society of Arrhythmia's Project. *Pacing and Clinical Electrophysiology* 34(8):1013–1027.

Quill, T., and Brody, H. 1996. Physician recommendations and patient autonomy: Finding a balance between physician power and patient choice. *Annals of Internal Medicine* 125(9):763–769.

Rapoport, M. 2013. Being a body or having one: automated domestic technologies and corporeality. *AI & Society* 28(2):209–218.

Sabanovic, S.; Bennett, C. C.; Chang, W.-L.; and Huber, L. 2013. PARO robot affects diverse interaction modalities in group sensory therapy for older adults with dementia. In *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*, 1–6. IEEE.

Trustworthiness and Safety for Intelligent Ethical Logical Agents via Interval Temporal Logic and Runtime Self-Checking

Stefania Costantini, Giovanni De Gasperis, Valentina Pitoni, Abeer Dyoub
Dip. di Ingegneria e Scienze dell'Informazione e Matematica (DISIM),
Università di L'Aquila, Coppito I-67100, L'Aquila, Italy
email: {stefania.costantini, giovanni.degasperis}@univaq.it, {valentina.pitoni, abeer.dyoub}@graduate.univaq.it

Abstract

Implementing Machine Ethics in Intelligent Agents involves trustworthiness and safety, meaning that agents should do what is expected they should do (at least, even in case of malfunctioning of any kind, concerning high-priority goals) and should *not* behave in unexpected potentially harmful ways. This topics are strongly related with "assurance", i.e., to ensuring that system users can rely upon the system. This paper deals with assurance of logical agent systems via temporal-logic-based runtime self-monitoring and checking.

Introduction

Intelligent Agents are at present and will be in the future more and more widely adopted in applications where living being or societal welfare can depend upon their behavior. In such application domains, agents' compliance to ethical principles is a mandatory requirement and the specific principles to be respected must be part of the system's specification. Thus, ensuring trustworthiness and safety of agent systems, in other words providing *assurance* of such systems, constitutes a crucial though difficult problem. In fact, agents represent a particularly complex case of dynamic, adaptive and reactive software systems. In Software Engineering, *certification* is aimed at producing evidence indicating that deploying a given system in a given application context involves the lowest possible level of risk (depending on the application at hand) of adverse consequences. Assurance, which has been defined as "the level of confidence that software is free from vulnerabilities, either intentionally designed into the software or accidentally inserted at any time during its lifecycle, and that the software functions in the intended manner" is related to dependability, i.e., to ensuring (or at least obtaining a reasonable confidence) that system designers and users can rely upon the system. An interesting discussion can be found in in (Winikoff 2010), and for the basic underlying concepts about verification and assurance of agent systems we invite the reader to refer to the relatively recent book (Dastani, Hindriks, and Meyer 2010) and to the references therein.

Pre-deployment (or "static" or "a priori") assurance and certification techniques for agent systems include verification and testing. We restrict ourselves to agent systems based upon computational logic, i.e., implemented in logic-based languages and architectures such as those presented in the survey (Bordini et al. 2006) and those mentioned in subsequent sections. Most verification methods for logical agents rely upon model-checking, and some (e.g., (Shapiro, Lesperance, and Levesque 2002)) upon theorem proving. Among recent interesting proposals about agent systems (pre-deployment) assurance we particularly mention (Winikoff 2010; 2017). About fault detection and recovery, a particularly interesting research work concerning model-checking is that of (Kouvaros and Lomuscio 2017) (see also the references therein).

It is widely acknowledged that industrial adoption of agents systems finds a serious obstacle in the stakeholders' lack of confidence about reliability of runtime behavior of such systems, even more so when the application domains involves moral or ethical requirements that must be fulfilled or at the very least should not be violated. As the applications of autonomous agents are inevitably increasing, and the adoption of such systems become more pervasive, the requirement that agents function in ethically responsible and safe manners becomes a pressing concern. Thus, in this paper we advocate methods for run-time monitoring and self-correction of agent systems, so as to enforce ethic behavior and to prevent violations. Citing (Rushby 2008), *. . . the use of adaptive systems for greater resilience create situations where runtime verification and monitoring could be particularly valuable. . . . Within suitable new frameworks, some of the evidence required for certification can be achieved by runtime monitoring - by analogy with runtime verification, this approach can, somewhat provocatively, be named "runtime certification"*. In fact, in our view the ultimate objective should be that of agents and agent systems *certified* to be ethically safe and secure.

The methods that we propose are not in alternative but rather complementary to the many existing verification and testing methodologies. For formalizing and implementing runtime self-checking in logical agents while coping with unanticipated circumstances, we propose temporal-logic-based special constraints to be dynamically checked at a certain frequency, customizable for each different constraint, depending upon how crucial is the requirement that the constraint is aimed to check. These constraints are based upon a simple interval temporal logic particularly tailored to the agent realm, A-ILTL ('Agent-Oriented Interval LTL', LTL standing as customary for 'Linear Temporal Logic'). The adoption of an interval temporal logics is motivated by the usefulness of being able to specify of time-bounded properties: namely, A-ILTL it makes it possible to specify that some property should occur within a certain time frame or before/after a certain time, where each interval can also be conditionally defined. A-ILTL constraints are conditional, i.e., they can be specified in a general form and each time they are checked they are instantiated (via suitable preconditions) to the present agent's state.

In (Rushby 2008), it is advocated that for adaptive systems (of which agents are clearly a particularly interesting case) assurance methodologies should whenever possible imply not only detection but also recovery from software failures. In fact, though (at least in principle) a certified software should not fail, in practice serious software-induced incidents have been observed in certified critical systems. In (Rushby 2008) examples are produced concerning airplane and air traffic control, where failures are often due on the one hand to incomplete specifications and on the other hand to unpredictability of the environment. Clearly, self-driving cars or eHealth systems actually in charge of patients (we have been experimenting on such systems, cf. (Aielli et al. 2016)) can incur in unwanted unanticipated situations that must be suitably coped-with. (Butner and Ghodoussi 2003), which discusses medical robotic applications in human telesurgery, emphasizes how critical systems should be designed so as to be *fail safe* in the sense that, in the event of failure, they proactively respond in order to limit harm to other devices or danger to users. We may notice that making a system fail safe is a part of ensuring the system's ethically correct behavior, in that such behavior should be preserved under any circumstances.

Our approach provides in fact the possibility of correcting and/or improving agent's behavior: the behavior can be corrected whenever an anomaly is detected, but can also be improved whenever it is acceptable, yet there is room for getting a better performance. Counter measures can be at the object-level, i.e., can be related to the application, or at the meta-level, e.g., replacing (as suggested in (Rushby 2008)) a software component by a diverse alternative. A-ILTL constraints are defined over formulas of any underlying logic language $\mathcal{L}$, and are rooted in the Evolutionary Semantics of agent programs (Costantini and Tocchio 2006): we have in fact integrated A-ILTL into this general semantic framework, that encompasses in a natural way many of the existing logic agent-oriented and languages. We thus obtain a fairly general setting, that can be adopted (as seen below) in several logic agent-oriented languages and formalisms.

This paper stems (concerning Evolutionary Semantics and A-ILTL) from (Costantini and Tocchio 2006; Costantini 2012; Costantini and De Gasperis 2013; 2014), where however (Costantini 2012; Costantini and De Gasperis 2013) appear on informal proceedings and (Costantini and De Gasperis 2014) appear only in the proceedings of a National Conference; this work has also been influenced by (Costantini et al. 2009; Costantini 2013). The application to Machine Ethics (principles and approach) is new, and to the best of our knowledge unprecedented in the literature. We have been stimulated and to some extent influenced by the important recent book by Luis Moniz Pereira on programming Machine Ethics (Pereira and Saptawijaya 2016).

The paper is organized as follows. In the first (Background) section we recall the Evolutionary Semantics of agent systems and the A-ILTL logic and constraints; this section can possibly be skipped by the non-expert in logic, as we have tried to make the rest of the paper self-contained and readable by means of intuitive explanations. Subsequently we introduce a case study to enlighten some kinds of ethical problems that intelligent agents might encounter. Then we illustrate (mostly by means of examples) how A-ILTL constraints can be exploited for runtime monitoring, self-checking and and self-repair of agent systems so as to cope with this kind of problems. Finally we discuss related work and propose some concluding remarks.

Background

Evolutionary Semantics

The Evolutionary semantics was introduced in (Costantini and Tocchio 2006) with the aim to provide a high-level general account of evolving agents, trying to abstract away from the details of specific agent-oriented frameworks. This is why we base the A-ILTL logic presented below upon such semantics. We in fact define, in very general terms, an agent as the tuple $Ag = < P_{\mathcal{A}}, E >$ where $\mathcal{A}$ is the agent name and $P_{\mathcal{A}}$ (that we call "agent program", but can be in turn a tuple) describes the agent according to some agent-oriented formalism $\mathcal{L}$. E is the set of the events that the agent is able to recognize or determine (so, E includes actions that the agent is able to perform).

Let H be the *history* of an agent as recorded by the agent itself, i.e., H includes agent's perceptions and *memories*. For instance, in the DALI agent-oriented language (Costantini and Tocchio Costantini and Tocchio2004) the history consists of: the set Ev of external and internal events, that represent respectively events that the agent presently perceives of its environment, and events that the agent has raised by its own internal reasoning processes; the set Act of the actions that the agent is enabled to perform at its present stage of operation; the set P of most recent versions "past events", which include: previously perceived events, but also actions that the agent has performed (notice that elements of Ev and Act will be transferred into P at the next stage); the set PNV of previous instances of past events (e.g., P may contain the last measurements of temperature while PNV may contain older ones), plus *past constraints* that specify interaction between P and PNV.

We assume that program $P_{\mathcal{A}}$ as written by the programmer is in general transformed into an initial agent program P_0 by means of an *initialization step*. When agent $\mathcal{A}$ is activated P_0 will go into execution, and will evolve according to events that either happen or are generated internally, to actions which are performed, etc. Thus, an agent evolves according to the evolution of its history H.

Evolution is formally represented via program-transformation steps, each one transforming P_i into

P_{i+1} according to H_i, which is the partial history up to stage i. The choice of which elements of H_i do actually trigger an evolution step is part of the definition of a specific agent framework.

Thus, we obtain a Program Evolution Sequence $PE = [P_0, \ldots, P_n, \ldots]$. The program evolution sequence will imply a corresponding Semantic Evolution Sequence $ME = [M_0, \ldots, M_n, \ldots]$ where M_i is the semantics of P_i according to $\mathcal{L}$, as the approach is parametric w.r.t $\mathcal{L}$. The evolutionary semantics ε^{Ag} of agent Ag is a tuple $\langle H, PE, ME \rangle$, where H is the history of Ag, and PE and ME are respectively its program and semantic evolution sequences.

Evolution is in principle of unlimited length, so the *snapshot at stage i* of ε_i^{Ag} is the tuple $\langle H_i, P_i, M_i \rangle$, where H_i is the history up to the events that have determined the transition from P_{i-1} to P_i. In (Costantini and Tocchio 2006), program transformation steps associated with DALI language constructs are defined in detail. They can easily be adapted to AgentSpeak (Rao 1996) as the two languages share a number of similarities. More generally however, in the specific agent setting under consideration an evolution step will occur at least whenever new events are perceived, reacted to, and recorded, and whenever an agent proactively undertakes measures to pursue its goals. An evolution step will possibly determine an update of the history, which is a part of the agent's *belief/knowledge base*. Thus, each evolution step affects the belief or "mental" state of an agent. The evolutionary semantics may express for instance the notion of *trace* of a GOAL agent (Hindriks, van der Hoek, and Meyer 2012) where agent program P_i encompasses the agent's *mental state* and each evolution step, which in GOAL is called *computation step* is determined by a *conditional action*. For 3APL (Dastani, van Riemsdijk, and Meyer 2005), agent program P_i encompasses the agent's *initial configuration*, and the related sets GR of *goal rules*, PR of *plan rules*, IR of *interactive rules*; the evolutionary semantics corresponds to a 3APL agent *computation run*, and evolution steps are determined by the 3APL *transition system*.

The adoption of Evolutionary Semantics in our approach is motivated by the assumption that *as agents are evolving entities whose behavior adapts to a changing and potentially unpredictable environment, ethic behavior must be checked and enforced along the entire agent's 'life'.*

A-ILTL

For defining properties that are supposed to be respected by an evolving system, a well-established approach is that of Temporal Logic, and in particular of Linear-time Temporal Logics (LTL). These logics are called 'linear' because (in contrast to 'branching time' logics) they evaluate each formula with respect to a vertex-labeled infinite path (or "state sequence") $s_0 s_1 \ldots$ where each vertex s_i in the path corresponds to a point in time (or "time instant" or "state"). In what follows, we use the standard notation for the best-known LTL operators.

An interval-based extension to the well-known linear temporal logic LTL is formally introduced in (Costantini 2012) where it is called A-ILTL for 'Agent-Oriented Interval LTL', which retains the underlying discrete linear model of time and the complexity of LTL. This simple formulation can thus be efficiently implemented, and is nevertheless sufficient for expressing and checking a number of interesting properties of agent systems.

A-ILTL expressions are (like plain LTL ones) interpreted in a discrete, linear model of time. Formally, this structure is represented by $\mathcal{M} = \langle \mathbb{N}, \mathcal{I} \rangle$ where, given countable set Σ of atomic propositions, interpretation function $\mathcal{I} : \mathbb{N} \mapsto 2^{\Sigma}$ maps each natural number i (representing state s_i) to a subset of Σ. Given set $\mathcal{F}$ of formulas built out of classical connectives and of LTL and A-ILTL operators (where however nesting of A-ILTL operators is not allowed), the semantics of a temporal formula is provided by a satisfaction relation: for $\varphi \in \mathcal{F}$ and $i \in \mathbb{N}$ we write $\mathcal{M}, i \models \varphi$ if, in the satisfaction relation, φ is true w.r.t. $\mathcal{M}, i$. We can also say (leaving $\mathcal{M}$ implicit) that φ *holds* at i, or equivalently in state s_i, or that state s_i satisfies φ. A structure $\mathcal{M} = \langle \mathbb{N}, \mathcal{I} \rangle$ is a model of φ if $\mathcal{M}, i \models \varphi$ for some $i \in \mathbb{N}$.

Some among the A-ILTL operators are the following, where we let $\varphi \in \mathcal{F}$ and m, n be positive integer numbers.

$F_{m,n}$ *(eventually (or "finally") in time interval)*. $F_{m,n}\varphi$ states that φ has to hold sometime on the path from state s_m to state s_n. I.e., $\mathcal{M}, i \models F_{m,n}\varphi$ if there exists j such that $j \geq m$ and $j \leq n$ and $\mathcal{M}, j \models \varphi$. Can be customized into F_m, *bounded eventually (or "finally")*, where φ should become true somewhere on the path from the current state to the (m)-th state after the current one.

$G_{m,n}$ (*always in time interval*). $G_{m,n}\varphi$ states that φ should become true at most at state s_m and then hold at least until state s_n. I.e., $\mathcal{M}, i \models G_{m,n}\varphi$ if for all j such that $j \geq m$ and $j \leq n$ $\mathcal{M}, j \models \varphi$. Can be customized into G_m, *bounded always*, where φ should become true at most at state s_m.

$N_{m,n}$ (*never in time interval*). $N_{m,n}\varphi$ states that φ should not be true in any state between s_m and s_n, i.e., $\mathcal{M}, i \models N_{m,n}\varphi$ if there not exists j such that $j \geq m$ and $j \leq n$ and $\mathcal{M}, j \models \varphi$.

A-ILTL and Evolutionary Semantics

We refine A-ILTL so as to operate on a sequence of states that corresponds to the Evolutionary Semantics stages defined before. In fact, states in our case are not simply intended as time instants: rather, they encompass stages of the agent evolution. *The connection that we establish here between A-ILTL and Evolutionary semantics is motivated from the fact that for enforcing ethical behavior it is in general necessary to inspect components of the agent's state, e.g., the present goals, the adopted plans, the actions to be executed, ect., as illustrated by the examples in the next sections.*

Time in this setting is considered to be local to the agent, where with some sort of "internal clock" is able to timestamp events and state changes. We borrow from (Henzinger, Manna, and Pnueli 1992) the following definition of *timed state sequence*, that we tailor to our setting.

If σ is a (finite or infinite) sequence of states, where the i-th state e_i, $e_i \geq 0$, is the *semantic snapshots at stage i* ε_i^{Ag} of given agent Ag, and T be a corresponding sequence of time instants t_i, $t_i \geq 0$, we have the following. A *timed* *state sequence for agent* Ag is the couple $\rho_{Ag} = (\sigma, T)$. Let ρ_i be the i-th state, $i \geq 0$, where $\rho_i = \langle e_i, t_i \rangle = \langle \varepsilon_i^{Ag}, t_i \rangle$.

We in particular consider timed state sequences which are *monotonic*, i.e., if $e_{i+1} \neq e_i$ then $t_{i+1} > t_i$. In our setting, it will always be the case that $e_{i+1} \neq e_i$ as there is no point in semantically considering a static situation: as mentioned, a transition from e_i to e_{i+1} will in fact occur when something happens, externally or internally, that affects the agent.

Then, in the above definition of A-ILTL operators, it is immediate to let $s_i = \rho_i$. This requires however a refinement: in fact, in a writing Op_m or $Op_{m,n}$ occurring in an agent program parameters m and n will not necessarily coincide with time instants of the above-defined timed state sequence. To fill this gap, in (Costantini 2012) a suitable approximation is introduced.

We need to adapt the interpretation function $\mathcal{I}$ of LTL to our setting. In fact, we intend to employ A-ILTL within logic-based agent-oriented languages for which an evolutionary semantics and a notion of logical consequence can be defined. Thus, given agent-oriented language $\mathcal{L}$ at hand, the set Σ of propositional letters used to define the A-ILTL semantic framework will coincide with all ground expressions of $\mathcal{L}$ (an expression is *ground* if it contains no variables, and each expression of $\mathcal{L}$ has a possibly infinite number of ground versions). A given agent program can be taken as standing for its (possibly infinite) ground version, as it is customarily done in many approaches. Notice that we have to distinguish between logical consequence in $\mathcal{L}$, that we indicate as $\models_{\mathcal{L}}$, from logical consequence in A-ILTL, indicated above simply as $\models$. However, the correspondence between the two notions can be quite simply stated by specifying that in each state s_i the propositional letters implied by the interpretation function $\mathcal{I}$ correspond to the logical consequences of agent program P_i:

Therefore, we let $\mathcal{L}$ be a logic language, $Expr_{\mathcal{L}}$ be the set of ground expressions that can be built from the alphabet of $\mathcal{L}$, ρ_{Ag} be a timed state sequence for agent Ag, and $\rho_i = \langle \varepsilon_i^{Ag}, t_i \rangle$ be the ith state, with $\varepsilon_i^{Ag} = \langle H_i, P_i, M_i \rangle$. Then, an A-ILTL formula τ is defined over sequence ρ_{Ag} if in its interpretation structure $\mathcal{M} = \langle \mathbb{N}, \mathcal{I} \rangle$, index $i \in \mathbb{N}$ refers to ρ_i, which means that $\Sigma = Expr_{\mathcal{L}}$ and $\mathcal{I} : \mathbb{N} \mapsto 2^{\Sigma}$ is defined such that, given $p \in \Sigma$, $p \in \mathcal{I}(i)$ iff $P_i \models_{\mathcal{L}} p$. Such an interpretation structure will be indicated with $\mathcal{M}^{Ag}$. We will thus say that τ holds/does not hold w.r.t. ρ_{Ag}.

A-ILTL properties are meant to be verified at run-time, and thus they act as *constraints* over the agent behavior (so, we will indifferently talk about A-ILTL rules, (meta-)axioms, expressions, or constraints). In an implementation, verification may not occur at every state (of the given interval). Rather, sometimes properties will be verified with a certain frequency, that can be specifically tuned to the various cases. To this aim, we have introduced a further extension that consists in defining subsequences of the sequence of all states: if *Op* is any of the operators introduced in A-ILTL and $k > 1$, Op^k is a semantic variation of *Op* where the sequence of states ρ_{Ag} of given agent is replaced by the subsequence $s_0, s_{k_1}, s_{k_2}, \ldots$ where for each $k_r, r \geq 1$, $k_r \ mod\ k = 0$, i.e., $k_r = g \times k$ for some $g \geq 1$.

A-ILTL formulas to be associated to given agent can be defined within the agent program, though they constitute an additional but separate layer, composed of formulas $\{\tau_1, \ldots, \tau_l\}$. Agent evolution must thus obey all these properties. precisely, given agent Ag and given a set of A-ILTL expressions $\mathcal{A} = \{\tau_1, \ldots, \tau_l\}$, timed state sequence ρ_{Ag} is *coherent* w.r.t. $\mathcal{A}$ if A-ILTL formula $G\zeta$ with $\zeta = \tau_1 \wedge \ldots \wedge \tau_n$ holds. Notice that the expression $G\zeta$ is an *invariance property* in the sense of (Manna and Pnueli 1984). In fact, coherence requires this property to hold for the whole agent's "life". In the formulation $G_{m,n}\zeta$ that A-ILTL allows for, one can express *temporally limited coherence*, concerning for instance "critical" parts of an agent's operation. Or also, one might express forms of *partial* coherence concerning only some properties.

An "ideal" agent will have a coherent evolution, whatever its interactions with the environment can be, i.e., whatever sequence of events arrives to the agent from the external "world". However, in practical situations such a favorable case will seldom be the case, unless static verification has been able to ensure total correctness of agent's behavior. Instead, violations will occasionally occur, and actions must be undertaken so as to attempt to regain coherence for the future. A-ILTL formulas in their practical form (seen below) encompass in fact the definition and execution of such actions.

A-ILTL Constraints for Self-Checking

In this section we illustrate how to define meta-level constraints constructed on the basis of A-ILTL formulas for defining and verifying liveness and safety properties in agent systems. Such verification can be particularly useful for ensuring properties related to machine ethics issues. We remind the reader that, in Software Engineering, *liveness* properties concern the progress that an agent makes and express that a (good) state eventually will be reached, while *safety* properties express that some (bad) state will never be entered. Thus, liveness is concerned with the evolution of a system, while in general safety is not: paradoxically, doing nothing prevents bad states from being reached. In our setting however we restricted ourselves to monotonic state sequences based upon the evolutionary semantics, so in agents evolve by definition. If violated, liveness properties are violated in infinite time (a good state not yet reached might be in principle reached in the future) while safety properties are violated in finite time, in case a "bad" state is reached. It is widely acknowledged that any property can be expressed as a conjunction of a safety and a liveness property. In agents, "bounded" liveness properties that can be expressed via A-ILTL are often more interesting than "pure" liveness: in fact, in many cases it does not suffice that a certain state might be reached in an indefinite future, as agents are situated real-time working entities that operate with limited computational resources and within deadlines. Bounded liveness properties are equivalent to safety properties that are violated whenever the desirable state is not reached withing the deadline.

A-ILTL formulas can be defined either on finite intervals and then, to any practical extent, they define safety properties, or on infinite intervals (with no upper bound) thus defining liveness properties. A-ILTL formulas can be employed to define special constraints, which are actually meta-axioms, that may constitute the *check layer* of an agent to many purposes, in particular to verify that the agent's behavior respects the parts of its specification concerning ethical behavior. The general form of a *Reactive A-ILTL constraint* (also called 'axiom', or 'rule') is the following (Costantini 2012) (where M, N, K can be either variables or constants):

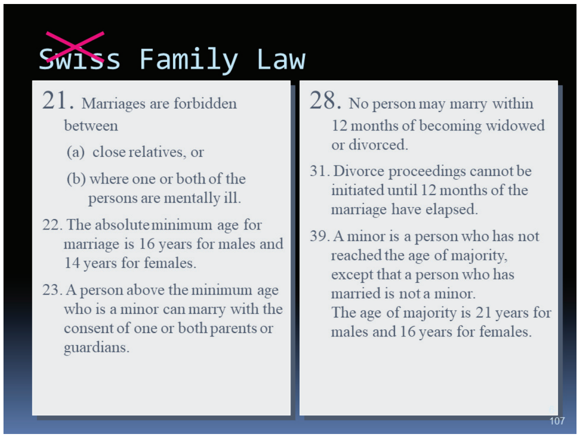
$$OP(M, N; K)\varphi :: \chi \div \rho$$

where:(i) $OP(M, N; K)\varphi :: \chi$ is an A-ILTL formula, called the *monitoring condition*, that in general involves the observation of either external or internal events; violation of the monitoring condition means that the agent's functioning is to some extent unsatisfactory with respect to the parts of its specification which are encoded in the condition itself. For instance, $EVENTUALLY(m, n; k)\,\varphi$ states that φ should become true at some point between time instants (states) m and n. (ii) ρ is called the *recovery component* of the rule, and it consists of a complex reactive pattern to be executed if the monitoring condition fails in order to restore an agent's acceptable behavior.

Thus, whenever the monitoring condition (automatically checked at frequency K) is violated (i.e., it does not hold) within given interval, then the recovery component ρ is executed. Frequency can be expressed in terms of states, or time instants. Setting frequency is very important, as it concerns how promptly a violation or fulfillment are detected, or a necessary measure is undertaken; specific K will depend on the kind of property one wants to check. Syntax and semantics of reactive patterns usable in the recovery component will depend upon the underlying language $\mathcal{L}$. In the examples, we adopt a sample syntax suitable for logic-programming-based settings: in fact, we mainly refer to agent-oriented rule-based programming languages like, e.g., GOAL, 3APL and DALI. For simplicity, under this assumption we restrict φ to be a conjunction of literals. φ must be ground when the formula is checked. However, similarly to negation-as-failure (where the negated atom can contain variables, that must be instantiated by literals evaluated previously), we allow variables to occur in an A-ILTL formula, to be instantiated via the execution of χ. Thus, from the procedural point of view, χ is required to be evaluated in the first place so as to make the A-ILTL formula ground. Notice that, for the evaluation of φ, χ and ρ, we rely upon the procedural semantics of the 'host' language. In (Costantini 2012) it is specified how to *operationally* perform such evaluation (how to check whether a formula holds).

A Case Study

Below we refer to a humorous though instructive case study proposed in an invited talk some years ago by Prof. Marek Sergot (Imperial College, London)[1]. As a premise, let us recall that, since 1600, ethics and morals relate to "right" and "wrong" conduct. Though these terms are sometimes used interchangeably, they are different: ethics refer to rules provided by an external source (typically by a social/cultural group), while morals refer to an individuals own principles regarding right and wrong: for instance, a lawyer's morals may tell her that murder is reprehensible and that murderers should be punished, but her ethics as a professional lawyer, require her to defend her client to the best of her abilities, even if she knows that the client is guilty. However, in the following we deliberately assume that immoral behavior can also be considered as unethical: though in general personal morality transcends cultural norms, is a subject of future debate if this can be the case for artificial agents.



~~Swiss~~ Family Law

21. Marriages are forbidden between
    (a) close relatives, or
    (b) where one or both of the persons are mentally ill.

22. The absolute minimum age for marriage is 16 years for males and 14 years for females.

23. A person above the minimum age who is a minor can marry with the consent of one or both parents or guardians.

28. No person may marry within 12 months of becoming widowed or divorced.

31. Divorce proceedings cannot be initiated until 12 months of the marriage have elapsed.

39. A minor is a person who has not reached the age of majority, except that a person who has married is not a minor.
The age of majority is 21 years for males and 16 years for females.

107

Figure 1: Case Study

The case study considers Romeo and Juliet who, as it is well-known, strongly wish to get married. As we will see, many plans are actually possible to achieve this goal (beyond getting killed or committing suicide like in Shakespeare's tragedy). Prior to executions, such plans should must be evaluated w.r.t. effectiveness, timeliness and feasibility, but and also w.r.t. deontic (ethical/moral and legal) notions. Prof. Sergot referred, due to its simplicity, to an excerpt of the Swiss Family Law reported in Figure1.

The problem for Romeo and Juliet is that they are both minors, and will never get their parents' consent to marry each other. Surprisingly enough, there are a number of feasible plans beyond waiting for reaching the majority age, among which:

(P1) Both Romeo and Juliet marry someone else, then divorce, and marry each other as married people acquire majority by definition; this plan requires a minimum of 24 months to be completed.

(P1') Variation of Plan 1 in case the spouse would not agree upon divorce: sleep with someone else, so as to force such agreement.

(P2) Both Romeo and Juliet marry someone else, then kill the spouses and marry each other; this plan is faster, as it takes a minimum of 12 months to be completed.

(P2') Variation of Plan 2 in case the act of killing constitutes a problem: hire a killer to do the job.

[1] Prof. Sergot kindly granted us via a personal communication the permission to report this example.

All the above plans are feasible, though some of them include actions which are generally considered as immoral, namely sleeping with someone else when married, and actions which are generally considered as unethical, namely killing someone or hiring a killer, where the latter ones are also illegal and imply a punishment. Notice that the possible plans might be different in case one referred to some other country; also what is illegal might change, for instance sleeping with someone else accounts to adultery which in many countries is punished; even divorce is not allowed everywhere. This if one implicitly refers to reality as the context of agent's activity. Instead, if one does not refer to reality but to some other context, e.g., to virtual storytelling or to a videogame, then every action assumes a different weight, as in playful contexts everything is allowed (except maybe for "serious" games with educational purposes).

So, we can draw at least the following indications from the case study:

- the context is relevant to moral/ethical/legal issues;
- some actions are not moral or non-ethical, and some of them are also illegal and lead to punishment;
- agents' plans to reach a goal should be evaluated "a priori" against including immoral/unethical/illegal actions;
- immoral/unethical/illegal actions should be prevented anyway, whenever they occur.

Marek Sergot made use of a concept of *counts as* (well-known in legal theory and other fields). For instance, *sleep with* (someone else than the spouse) counts as *adultery*, which is an *institutional* concept considered as immoral and potentially also illegal, and *kill* counts (not always but in many situations, including that of the example) as *murder*, another institutional concept normally considered as both unethical and illegal.

Notice that the above aspects relate to safety properties that should be enforced, that can be rephrased as follows:

- never operate w.r.t. an incorrect context (the information about the present context must always be up-to-date);
- never execute actions that are deemed not acceptable (immoral/unethical/illegal) in the present context, and never execute plans including such actions.

Another aspect that we emphasize is that of *commitment*: Romeo and Juliet are committed to marry each other, and will for no reason give up this intention. In the theory of rational agents, and in particular of BDI agents (Rao and Georgeff 1991), commitment to an intention (i.e., a desire which has been adopted as an actual goal) can be of three kinds of strength: (i) *blind* commitment, where an agent never gives up a goal, whatever the circumstances, until it is reached; (ii) *single-minded* commitment, where a goal is pursued until reached or no longer believed possible; (iii) *open-minded* commitment, where a goal can be opportunistically dropped if a more desirable option arises. Clearly, opportunism generally clashes with ethical issues, so ethical agents will pursue their assigned tasks according to (i) or (ii), depending upon the kind of task (the case is different for activities which do not involve moral/ethical aspects).

Implementation of the Case Study: Sketch

In order to demonstrate the potential usefulness of runtime self-cheking and correction in enforcing/verifying agents' ethical behavior we discuss excerpts from a possible implementation of the case study, in order to provide a general idea. Let us assume to add to the language a transitive predicate $\mathit{COUNTS\ AS}$ which is used (in infix form) in expressions of the form exemplified below.

$\mathit{kills}\ \mathit{COUNTS\ AS}\ \mathit{murder}\ \mathit{CONDS}\ \ldots$

where after CONDS we have the (optional) conditions under which $\mathit{COUNTS\ AS}$ applies: concerning the case study, they define in which cases killing accounts to murder (e.g., it was no self-defense, it does not occur during a battle in war, etc.). Such statements are related to the present context so for the case study, and assuming to deal with a real situation under European legislation, we might also have:

$\mathit{sleep\_with}\ \mathit{COUNTS\ AS}\ \mathit{adultery}$
$\mathit{adultery}\ \mathit{COUNTS\ AS}\ \mathit{immoral}$
$\mathit{adultery}\ \mathit{COUNTS\ AS}\ \mathit{unethical}$
$\mathit{murder}\ \mathit{COUNTS\ AS}\ \mathit{unethical}$
$\mathit{adultery}\ \mathit{COUNTS\ AS}\ \mathit{illegal}$

The formalization will in general include general statements such as for instance the following, that state that an ethical agent will never give up the tasks to which it is committed, and will never violate either the law or generally-accepted rules of behavior, wherever commitment or behavior-related violation might lead to direct or indirect harm to humans, animals, other agents, etc.:

$\mathit{violate\_commitment}\ \mathit{COUNTS\ AS}\ \mathit{unethical}$
$\mathit{violate\_law}\ \mathit{COUNTS\ AS}\ \mathit{unethical}$
$\mathit{improper\_behavior}\ \mathit{COUNTS\ AS}\ \mathit{unethical}$

For example, for a human car driver, some rules to follow are (among others) to respect the traffic laws and to avoid behavior that may lead to harm for others or to damage to public property. So we may have:

$\mathit{talk\_on\_the\_phone}\ \mathit{COUNTS\ AS}\ \mathit{dangerous\_driving}$
$\mathit{high\_speed}\ \mathit{COUNTS\ AS}\ \mathit{dangerous\_driving}$
$\mathit{ignore\_pedestrians}\ \mathit{COUNTS\ AS}\ \mathit{dangerous\_driving}$
$\mathit{step\_over\_grass}\ \mathit{COUNTS\ AS}$
$\qquad\qquad \mathit{damage\_public\_property}$
$\ldots$
$\mathit{dangerous\_driving}\ \mathit{COUNTS\ AS}$
$\qquad\qquad \mathit{violate\_commitment}$
$\mathit{dangerous\_driving}\ \mathit{COUNTS\ AS}$
$\qquad\qquad \mathit{improper\_behavior}$
$\mathit{damage\_public\_property}\ \mathit{COUNTS\ AS}$
$\qquad\qquad \mathit{violate\_law}$
$\ldots$

Below we show some A-ILTL rules/constraints useful in the formalization of the case study. First of all, we introduce an A-ILTL rule for context change:

$$\mathit{ALWAYS}\ \ \mathit{context\_change}(C, C_1) \div \\ \mathit{discharge\_context}(C),\ \mathit{assume\_context}(C_1)$$

In particular, whenever the agent perceives a change of context (e.g., the agent stops working and starts a videogame, or vice versa, or finishes a videogame and goes to help children with their homework) then all the relevant ethic assumptions (among which, for instance, the $COUNTS\ AS$ facts) about the new context C_1 must be loaded, while those relative to the previous context C must be dismissed; this is important because, e.g., after finishing a videogame it is no longer allowed to kill any living being in sight just for fun. . . Frequency of check of this constraint is not specified here, however it should guarantee a prompt enough adaptation to a change.

Then, we show an A-ILTL meta-axiom that usefully employs $COUNTS\ AS$ facts, that are either explicit or implicitly derived by transitivity (we do not enter in the detail of how to implement transitivity; suffices to say that this is possibly done, e.g., via other meta-level axioms). In runtime self-checking, as discussed above, an issue of particular importance in case of violation of a property is that of undertaking suitable measures in order to recover or at least mitigate the critical situation. Measures to be undertaken in such circumstances can be seen as an internal reaction. In particular, given now the present context for granted, the A-ILTL constraint below prevents any plan from being executed that includes even a single action which counts as unethical in the present context. Such constraint must be checked at suitable frequency (omitted here), so as to check all the plans that an agent may devise:

$$
\begin{array}{l}
ALWAYS \\
\quad goal(G), plan(G, P), element(Action, P) :: \\
\qquad Action\ COUNTS\ AS\ unethical \div \\
\qquad\qquad\qquad block\_plan\_execution(P)
\end{array}
$$

However, in case a plan is blocked the original goal G remains unfulfilled. The next A-ILTL axiom is a meta-statement expressing the capability of an agent to modify its own behavior to cope with such a situation: if a goal G which is crucial to the agent, possibly for its ethical behavior (e.g., providing a doctor or an ambulance to a patient in need), has not been achieved (in a certain context) and the initially allotted time has elapsed, then the recovery component implies replacing the planning module (assuming that more than one is available) and retrying the goal. We suppose that the *possibility* of achieving a goal G is evaluated w.r.t. a module M that represents the preconditions for G (notation $P(G, M)$, P standing for 'possible'). Necessity and possibility evaluation within a reasonably expressive framework has been discussed in (Costantini 2011). In case the goal is still deemed to be possible, the reaction/countermeasure consists in substituting the present planning module with another one and re-trying the goal.

$$
\begin{array}{l}
NEVER\ goal(G), \\
\quad crucial(G), \\
\quad timed\_out(G), not\ achieved(G), \\
\quad eval\_context(G, M), P(G, M) \div \\
\qquad\qquad replace\_planning\_module, retry(G)
\end{array}
$$

Time intervals (as allowed by A-ILTL definition) have never been exploited in the above examples. They can however be useful in many cases for the punctual definition of moral/ethical specific behaviors, e.g., never leave a patient or a child alone at night, and the like.

It is important to notice that the above sample meta-axioms access aspects of an agent's operation such as goals, plans, action execution, etc. This is made technically possible by the connection to the Evolutionary Semantics. In conceptual terms, *A-ILTL expression that enforce ethical properties exhibit a reflective/introspective behavior (Costantini 2002; Barklund et al. 2000) as they make an agent observe, inspect, evaluate, correct its own behavior. This in our view is by no means fortuitous: in fact, any 'animated' being that tries to be ethical must confront the 'instinctive' or random behavior to the underlying moral/ethical principles, and correct such behavior accordingly*.

Related Work and Concluding Remarks

In this paper we have proposed runtime constraints for agents' self-checking and monitoring in the perspective of implementing machine-ethics principles. We have shown how to express liveness and safety properties that can be useful to enforce at run-time ethical behavior in agents and to detect violations, also considering the different contexts an agent might be involved into. We have provided a general semantics, so as to allow such constraints to be adopted in different agent-oriented frameworks.

There are similarities between A-ILTL constraints and event-calculus formulations (Kowalski and Sergot 1986), and with approaches based on abductive logic programming such as SCIFF (cf. (Montali et al. 2011) and the references therein) and Reactive Event Calculus, which stems from SCIFF (Bragaglia et al. 2012); such approaches however have never been applied to Machine Ethics, and have been devised for static or dynamic checking performed by a third party. The use of temporal logic to define run-time monitors is discussed in (Barringer, Rydeheard, and Havelund 2010) and the references therein. However, this work is not related to agents and does not concern self-checking and recovery.

The complexity of checking A-ILTL expressions is discussed in (Costantini and De Gasperis 2014) where it is noted that though such complexity is relatively low, in order to avoid an excessive computational burden a designer should keep the number of A-ILTL expressions as limited as possible, and tune frequencies carefully. Our approach has been prototypically implemented, and has been experimented in energy management for smart buildings (Caianiello et al. 2013). Future work includes enhancing A-ILTL constraints to make them adaptive to different conditions. As suggested in (Rushby 2008), an interesting line of investigation concerns automated synthesis of runtime constraints from specifications.

References

Aielli, F.; Ancona, D.; Caianiello, P.; Costantini, S.; De Gasperis, G.; Marco, A. D.; Ferrando, A.; and Mascardi, V. 2016. FRIENDLY & KIND with your health: Human-friendly knowledge-intensive dynamic systems for the e-health domain. In volume 616 of *Communications in Computer and Information Science*, 15–26. Springer.

Barklund, J.; Dell'Acqua, P.; Costantini, S.; and Lanzarone,

G. A. 2000. Reflection principles in computational logic. *J. Log. Comput.* 10(6):743–786.

Barringer, H.; Rydeheard, D. E.; and Havelund, K. 2010. Rule systems for run-time monitoring: from eagle to ruler. *J. Log. Comput.* 20(3):675–706.

Bordini, R. H.; Braubach, L.; Dastani, M.; Fallah-Seghrouchni, A. E.; Gómez-Sanz, J. J.; Leite, J.; O'Hare, G. M. P.; Pokahr, A.; and Ricci, A. 2006. A survey of programming languages and platforms for multi-agent systems. *Informatica (Slovenia)* 30(1):33–44.

Bragaglia, S.; Chesani, F.; Mello, P.; Montali, M.; and Torroni, P. 2012. Reactive event calculus for monitoring global computing applications. In volume 7360 of *Lecture Notes in Computer Science*, 123–146. Springer.

Butner, S., and Ghodoussi, M. 2003. Transforming a surgical robot for human telesurgery. *IEEE Transactions on Robotics and Automation* 19(5):818–824.

Caianiello, P.; Costantini, S.; De Gasperis, G.; Florio, N.; and Gobbo, F. 2013. Application of hybrid agents to smart energy management of a prosumer node. In volume 217 of *Advances in Intelligent and Soft Computing*, 597–607. Springer.

Costantini, S., and De Gasperis, G. 2013. Meta-level constraints for complex event processing in logical agents. In *Informal Proc. of Commonsense 2013, 11th International Symposium on Logical Formalizations of Commonsense Reasoning*.

Costantini, S., and De Gasperis, G. 2014. Runtime self-checking via temporal (meta-)axioms for assurance of logical agent systems. In *Proceedings of the 29th Italian Conference on Computational Logic CILC 2014*, volume 1195 of *CEUR Workshop Proceedings*, 241–255. CEUR-WS.org.

Costantini, S., and Tocchio, A. 2006. About declarative semantics of logic-based agent languages. In *Declarative Agent Languages and Technologies III, Third International Workshop, DALT 2005, Selected and Revised Papers*, volume 3904 of *LNAI*. Springer. 106–123.

Costantini, S.; Dell'Acqua, P.; Pereira, L. M.; and Tsintza, P. 2009. Runtime verification of agent properties. In *Proc. of the Int. Conf. on Applications of Declarative Programming and Knowledge Management (INAP09)*.

Costantini, S. 2002. Meta-reasoning: A survey. In volume 2408 of *Lecture Notes in Computer Science*, 253–288. Springer.

Costantini, S. 2011. Answer set modules for logical agents. In *Datalog Reloaded: First International Workshop, Datalog 2010*, volume 6702 of *LNCS*. Springer. Revised selected papers.

Costantini, S. 2012. Self-checking logical agents. In *Proc. of LA-NMR 2012*, volume 911. CEUR Workshop Proceedings (CEUR-WS.org). Invited paper.

Costantini, S. 2013. Self-checking logical agents. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, Proceedings*, 1329–1330. IFAAMAS.

Costantini, S. and Tocchio, A. 2004 The DALI logic programming agent-oriented language. In *Proc. of JELIA-04*, volume 3229 of *Lecture Notes in Artificial Intelligence*, 685-688. Springer.

Dastani, M. 2015. Programming multi-agent systems. *Knowledge Eng. Review* 4: 394–418.

Dastani, M.; Hindriks, K V.; and Meyer, J-J Ch. (Eds.) 2010. Specification and Verification of Multi-agent Systems. Springer.

Dastani, M.; van Riemsdijk, B.; and Meyer, J-J Ch. Programming Multi-Agent Systems in 3APL. 2015. volume 15 of Multiagent Systems, Artificial Societies, and Simulated Organizations, 39-67. Springer.

Henzinger, T. A.; Manna, Z.; and Pnueli, A. 1992. Timed transition systems. In de Bakker, J. W.; Huizing, C.; de Roever, W. P.; and Rozenberg, G., eds., *Real-Time: Theory in Practice, REX Workshop, Proceedings*, volume 600 of *Lecture Notes in Computer Science*, 226–251. Springer.

Hindriks, K V.; van der Hoek. W.; and Meyer, J-J Ch. 2012. GOAL Agents Instantiate Intention Logic. In volume 7360 of *Lecture Notes in Computer Science*, 196–219. Springer.

Kouvaros, P., and Lomuscio, A. 2017. Verifying fault-tolerance in parameterised multi-agent systems. In *Proceedings IJCAI 2017*, 288–294.

Kowalski, R., and Sergot, M. 1986. A logic-based calculus of events. *New Generation Computing* 4:67–95.

Manna, Z., and Pnueli, A. 1984. Adequate proof principles for invariance and liveness properties of concurrent programs. *Sci. Comput. Program.* 4(3):257–289.

Montali, M.; Chesani, F.; Mello, P.; and Torroni, P. 2011. Modeling and verifying business processes and choreographies through the abductive proof procedure sciff and its extensions. *Intelligenza Artificiale, Intl. J. of the Italian Association AI*IA* 5(1).

Pereira, L. M., and Saptawijaya, A. 2016. *Programming Machine Ethics*, volume 26 of *Studies in Applied Philosophy, Epistemology and Rational Ethics*. Springer.

Rao, A. S. AgentSpeak(L): BDI Agents Speak Out in a Logical Computable Language. 1996. In volume 1038 of *Lecture Notes in Computer Science*, 42–55. Springer.

Rao, A. S., and Georgeff, M. 1991. Modeling rational agents within a BDI architecture. In *Proc. of the Second Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'91)*, 473–484. Morgan Kaufmann.

Rushby, J. M. 2008. Runtime certification. In *Runtime Verification, 8th International Workshop, RV 2008. Selected Papers*, volume 5289 of *Lecture Notes in Computer Science*. Springer. 21–35.

Shapiro, S.; Lesperance, Y.; and Levesque, H. 2002. The cognitive agents specification language and verification environment. In *Proceedings of AAMAS 2002*, 19–26.

Winikoff, M. 2010. Assurance of agent systems: What role should formal verification play? In *Specification and Verification of Multi-agent Systems*, 353–383. Springer.

Winikoff, M. 2017. BDI agent testability revisited. *Autonomous Agents and Multi-Agent Systems* 31(5):1094–1132.

Ethical Considerations for AI Researchers

Kyle Dent
Palo Alto Research Center
kdent@parc.com

Abstract

Use of artificial intelligence is growing and expanding into applications that impact people's lives. People trust their technology without really understanding it or its limitations. There is the potential for harm and we are already seeing examples of that in the world. AI researchers have an obligation to consider the impact of intelligent applications they work on. While the ethics of AI is not clear-cut, there are guidelines we can consider to minimize the harm we might introduce.

Introduction

A quick scan of recent papers covering the area of AI and ethics reveals researchers' admirable impulse to think about teaching intelligent agents human values (Abel, MacGlashan, and Littman 2016; Burton, Goldsmith, and Mattei 2016; Riedl and Harrison 2016). There is, however, another important and more immediate aspect of AI and ethics we ought to take into consideration. AI is being widely deployed for new applications; it's becoming pervasive; and it's having an effect on people's lives. AI researchers should reflect on their own personal responsibility with regard to the work they do. Many of us are motivated by the idea that we can contribute useful new technology that has a positive impact on the world. Positive outcomes have largely been the case with advanced technologies that improve cancer diagnosis and provide safety features in cars, for example. With vast amounts of computing power and a number of improved techniques, intelligent software is being adopted in more and more contexts that affect people's lives. How people use it is starting to matter, and the impact of our decisions matters.

Not surprisingly as the use of AI expands, negative consequences of its failures and design flaws are more visible. Much of the AI that has recently been deployed derives its intelligence from learning algorithms that are based on statistical analysis of data. The acquisition, applicability, and analysis of that data determine its output. Statistics shine when making predictions about distributions over populations. That predictive power fades when applied to individuals. There will be faulty predictions. The popular press is rife with misuses of statistical analysis and AI (Crawford 2016; O'Neil 2016). Given the growing use, the built-in uncertainties, and the public's tendency to blindly trust technology, we have a responsibility to consider the likely and unlikely outcomes of the choices we make when we are designing and developing tools or predictive systems to support decision making that affect people and communities of people.

Purposely malicious choices are obviously ethically unacceptable. In (Yampolskiy 2015), the author outlines various pathways that lead to dangerous artificial intelligence. Within the taxonomy, there are pathways that introduce danger into artificial intelligence 'on purpose.' The other pathways inadvertently lead to hazards in the system. You can decide for yourself if you are comfortable developing smart weapons, for example, and most of us would, at a minimum, pause to consider the implications of that decision. But the inadvertent pathways leading to dangerous AI can be difficult to foresee and may come about from subtle interactions. Our less obvious responsibility lies in giving careful consideration to our choices and being clear to ourselves and our stakeholders about assumptions, trade-offs, and choices we make.

Several other papers consider another ethical aspect in the fairness of automatic systems (O'Neil 2016; Hardt, Price, and Srebro 2016; NSTCCT 2016), and some even conclude that it's inherently impossible for most problems (Kleinberg, Mullainathan, and Raghavan 2016). One of the points I'll make is that discussions about fairness and societal impact can be cut off once an intelligent agent is introduced into the process. There is a popular feeling that machines don't make value judgments and are inherently unbiased. However, the assumptions we make when designing our systems are often based on subjective value judgments; for example, choosing data sets, selecting weighting schemes, balancing precision and recall. We have to be transparent about what we do and be clear about the choices we have made. The ultimate purpose matters and the decisions you come to must be communicated.

Blind Trust in Technology

Although there are pockets of skepticism towards intelligent systems, by and large people are content to offload decisions to technology. In May 2016, there was a widely publicized crash involving a Tesla Motors car being driven in computer-assisted mode. It appears the driver had undue faith in the capabilities of the car (Habib 2017). The following week another driver following a GPS unit steered her car into Ontario's Georgian Bay (MiQuigge 2016). These extreme examples reveal a trend in the general population to trust the smart devices in our lives.

Ideally government agencies and jurisdictions would apply the principles of open government and transparency when contracting with suppliers for decision-making tools. In practice that hasn't been the case. Last year, two researchers filed 42 open records requests in 23 different states asking for information about software with predictive algorithms used by governments as decision support tools (Brauneis and Goodman 2017). Their goal was to understand the policies built into the algorithms in order to evaluate their usefulness and fairness. Only one of the jurisdictions was able to provide information about the algorithms the software used and how it was developed. Some of those who did not respond cited agreements with vendors preventing them from revealing information, but many did not seem concerned about transparency in their process nor the need to understand the technology. Assuming the best intentions of the decision makers, they are also demonstrating great faith in the technology and vendors they contract with.

There is also evidence that users of these systems, judges and hiring managers for example, weight AI guidance too heavily. Without tools, when people are making decisions, there is public awareness that decisions are made within some context. We understand that individuals can be influenced even subconsciously by their biases and prejudices. Technologically assisted decisions tend to shut down the conversation about fairness despite their having a large effect on people's lives. Those affected may not have the opportunity to contest the decisions. If important decisions are made through our models, we must use care in developing them and clearly communicate the assumptions we make.

Ethical Obligations

Physicians and attorneys have well-established codes of ethics. Doctors famously commit to not doing any harm. Implied in that concept is the idea that there is potential to do harm. It is clear from many examples, some of which I mention in this paper, that there is the potential for harm in our work, and given people's lack of understanding of the limits of and the trust they place in technology, AI researchers have a personal, ethical obligation to reflect on the decisions we make.

Ethical thinking helps us to make choices and just as importantly provides a framework to reason about those choices. The framework we use (explicitly or not) is defined by a set of principles that guide and support our decisions. One of the difficult things about defining ethical standards is deciding the values to base them on. Ethics issues will undoubtedly be discussed and argued within the community and the world generally in the coming years. Each of us can start by considering our own roles and being consciously aware of the effects our work can have.

The stakeholders who decide to deploy intelligent decision making, government agencies for example, generally aren't qualified to assess the assumptions, models and algorithms in it. This asymmetrical relationship puts the burden on those with the information to be clear, honest, and forthcoming with it. Those at a disadvantage depend on us to inform them about technology's fitness for their purpose, its reliability and accuracy. We usually focus on the technical aspects of our work like selecting highly predictive models and minimizing error functions, but when applying algorithmic decision-making that will affect human beings, we have a responsibility to think about more.

Recommendations for Consideration

Ethics is not science. But it is possible to ground our thinking in well-defined guidelines to assist in making ethical decisions for AI development. A formal framework may even emerge within the researcher community with time. In the short-term, the following is a list of thoughts and questions to ask ourselves when designing predictive or decision-making systems.

1. Relevance of data and models

It is important to think carefully about the data used to train our technology. Are the data and models appropriate to the real-life problem they are solving? It is tempting to believe causal forces are at play when we find correlation on a single dataset. Does the data capture the true variable of interest? Is it consistent across observations and over time? We often introduce a proxy variable because the variable we need isn't available or isn't easy to quantify. Can your findings be calibrated against the real-world situation? Even better could you measure the actual outcome you're trying to achieve?

In 2008, Researchers at Google had the idea that an increase in search queries related to the flu and flu symptoms could be indicative of a spreading virus. They created the Google Flu Trends (GFT) web service to track Google users' search queries related to the flu. If they detected increased transmission before the numbers from the U.S. Centers for Disease Control and Prevention (CDC) came out, earlier interventions could reduce the impact of the virus. The initial article reported 97% accuracy using the CDC data as the gold standard (Ginsberg et al. 2009). However, a follow-up report showed that in subsequent flu seasons GFT predicted more than double what the CDC data showed (Lazer et al. 2014). Given the first year's high accuracy, it would have been easy for the researchers to believe they had discovered a strong, predictive signal. But online behavior isn't necessarily a reflection of the real world. There are several factors that might make the GFT data wrong. One of them is that the underlying algorithms of Google Search itself (the GFT researchers don't control those) can change from one year to the next. Also users' search behavior could have changed. Mainly, however, people's search patterns are probably not a good single indicator of a spreading virus. There are many other factors and various reasons people might search for information.

Training data rarely aligns with real-life goals. In (Lipton 2016), Lipton presents challenges to providing and even defining interpretability of machine learning outputs.

He identifies several possible points of divergence between training data and real-life situations. For example, off-line training data is not always representative of the true environment, and real-world objectives can be difficult to encode as simple value functions. Often we work with data that was collected for other purposes and almost never under ideal, controlled circumstances. What was the original purpose in collecting the data, and how did that determine its content? In July of 2015, another group at Google had to apologize for its Photos application identifying a black couple as gorillas (Guynn 2015). Their training dataset was not representative of the population it was meant to predict. Also, there are limits to the amount of generalization we can expect from any learning method trained on a particular dataset.

Is it possible your dataset contains biases? When making decisions related to hiring, judicial proceedings, and job performance, for example, many personal characteristics are legally excluded. Also, humans are good at discarding variables they recognize as irrelevant to the decision to be made; computers are blind to those considerations. Are there other characteristics that are closely correlated with legally and ethically protected ones? If you don't consider those, you can inadvertently treat people unfairly based on protected or irrelevant characteristics. There is often a trade-off between accuracy and the intelligibility of a model (Caruana et al. 2015). More predictive but harder-to-understand models can make it difficult to know which personal characteristics determine the decision and are therefore not available for validation against human judgment.

In (Caruana et al. 2015) the authors describe a system that learned a rule that patients with a history of asthma have a lower risk of dying from pneumonia. Based on the data used to train the system, their model was absolutely correct. However, in reality asthma sufferers (without treatment) have a higher risk of dying from pneumonia. Because of the increased risk, when patients with a history of asthma go to the hospital, the general practice is to place them in an intensive care unit. The extra attention they receive decreases their risk of dying from pneumonia even below that of the general population. It is our natural inclination to develop models with the highest accuracy. However, the necessity of visibility into decisions where people's lives are concerned, may increase the importance of explainability at the expense of some predictive performance. In all cases, our stakeholders must understand the decisions we make and the trade-offs implied by them.

2. Safeguards for Failures and Misuse

Even experienced researchers with the best intentions are inclined to favor the positive outcomes of their work. We highlight positive results, but we should also think through failure modes and possible unintended consequences. What about misuse? There isn't a lot you can do about a person determined to use the technology in ways it wasn't intended, but are there ways a good-faith user might go wrong? Can you add protections for that?

The 2016 Tesla accident mentioned before was catastrophic. The driver used computer-assisted mode in conditions it was expressly not designed for resulting in his death. The accident was investigated by two government agencies. The first finding from the National Highway Traffic and Safety Administration found that the driver-assist software had no safety defects and declared that, in general, the vehicles performed as designed (Habib 2017) implying that responsibility for use of the system falls on the operator. A later investigation from The National Transportation and Safety Board found otherwise (NTSB 2016). They declared that the automatic controls played a major role in the crash. The fact that the driver was able to use computer assistance in a situation it was not intended for was problematic. The combination of human error and insufficient safeguards resulted in an accident that should not have happened.

3. Accuracy

How accurate is your algorithm and how accurate does it need to be? Do your stakeholders understand the number of people who will be subject to a missed prediction given your measure of accuracy? A model that misses only 1% shows phenomenally good performance, but if hundreds or thousands of people are still adversely affected, that might not be acceptable. Are there human inputs that can compensate for the system's misses and can you design for that? What about post-deployment accuracy? Accuracy in training data doesn't always reflect real usage. Do you have a way to measure runtime accuracy? The world is dynamic and changes with time. Is there a way to continue to assess the accuracy after release? How often does it have to be reviewed?

4. Size and severity of impact

Think about the numbers of people affected. Of course, you want to avoid harming anyone but knowing the size or the severity of negative consequences can justify the cost of extra scrutiny. You might also be able to design methods that mitigate for them. Given an understanding of the impact, you can make better decisions about the value required by the extra effort.

Conclusion

Individual researchers, especially in commercial operations, don't always have the chance to communicate clearly and transparently with clients. At least being transparent with your immediate stakeholders can set the right expectations for them when they represent your work down the line. You are necessarily making decisions about the models and software you develop. If you don't surface those decisions to discuss their effect, they may never be brought to light.

A short paper cannot cover such a large and multi-faceted issue. The main idea is for each of us to think individually about our own responsibilities and the impact our work can have on real lives. It's useful to spend time thinking about our assumptions and the trade-offs we make in the context of the people who will be affected. Communicating those to everyone concerned is also critical. Modern versions of the Hippocratic Oath are still used by many medical schools. The spirit of the oath is applicable to most research affecting human beings. One phrase is especially general and worth keeping in mind:

> "I will remember that I remain a member of society, with special obligations to all my fellow human beings…" (Tyson 2001)

References

Abel, D.; MacGlashan, J.; and Littman, M. 2016. Reinforcement learning as a framework for ethical decision making. AI, Ethics, and Society Workshop at the Thirtieth AAAI Conference on Artificial Intelligence. Association for the Advancement of Artificial Intelligence.

Brauneis, R., and Goodman, E. P. 2017. Algorithmic transparency for the smart city. *Yale Journal of Law & Technology; GWU Law School Public Law Research Paper; GWU Legal Studies Research Paper*.

Burton, E.; Goldsmith, J.; and Mattei, N. 2016. Using "the machine stops" for teaching ethics in artificial intelligence and computer science. AI, Ethics, and Society Workshop at the Thirtieth AAAI Conference on Artificial Intelligence. Association for the Advancement of Artificial Intelligence.

Caruana, R.; Lou, Y.; Gehrke, J.; Koch, P.; Sturm, M.; and Elhadad, N. 2015. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, 1721–1730. New York, NY, USA: ACM.

Crawford, K. 2016. Artificial intelligence's white guy problem. *New York Times – June 25, 2016*.

Ginsberg, J.; Mohebbi, M.; Patel, R.; Brammer, L.; Smolinski, M.; and Brilliant, L. 2009. Detecting influenza epidemics using search engine query data. *Nature* 457:1012–1014. doi:10.1038/nature07634.

Guynn, J. 2015. Google photos labeled black people 'gorillas'. *USA Today*. Online; posted July 1, 2015.

Habib, K. 2017. Pe 16-007 automatic vehicle control systems. Technical report, National Highway Traffic Safety Administration.

Hardt, M.; Price, E.; and Srebro, N. 2016. Equality of opportunity in supervised learning. *CoRR* abs/1610.02413.

Kleinberg, J. M.; Mullainathan, S.; and Raghavan, M. 2016. Inherent trade-offs in the fair determination of risk scores. *CoRR* abs/1609.05807.

Lazer, D.; Kennedy, R.; King, G.; and Vespignani, A. 2014. The parable of google flu: Traps in big data analysis. *Science* 343(14 March):1203–1205.

Lipton, Z. C. 2016. The mythos of model interpretability. *CoRR* abs/1606.03490.

MiQuigge, M. 2016. Woman follows gps; ends up in ontario lake. *Toronto Sun – May 13, 2016*.

NSTCCT. 2016. Preparing for the future of artificial intelligence. Technical report, National Science and Technology Council Committee on Technology.

NTSB. 2016. Highway accident report: Collision between a car operating with automated vehicle control systems and a tractor-semitrailer truck. Technical report, National Transportation Safety Board.

O'Neil, C. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. New York, NY, USA: Crown Publishing Group.

Riedl, M., and Harrison, B. 2016. Using stories to teach human values to artificial agents. AI, Ethics, and Society Workshop at the Thirtieth AAAI Conference on Artificial Intelligence. Association for the Advancement of Artificial Intelligence.

Tyson, P. 2001. The hippocratic oath today. *PBS - NOVA*. Online; posted March 3, 2001.

Yampolskiy, R. V. 2015. Taxonomy of pathways to dangerous AI. *CoRR* abs/1511.03246.

Interactive Agent that Understands the User

Piotr Gmytrasiewicz
Department of Computer Science
University of Illinois at Chicago
851 S. Morgan St.
Chicago, IL 60607
piotr@uic.edu

George Moe
Computer Science Department
Harvard University
Cambridge, MA 02138
geomo3@gmail.com

Adolfo Moreno
Department of Computer Science
University of Illinois at Chicago
851 S. Morgan St.
Chicago, IL 60607
asmoren2@uic.edu

Abstract

Our work uses the notion of theory of mind to enable an interactive agent to keep track of the state of knowledge, goals and intentions of the human user, and to engage in and initiate sophisticated interactive behaviors using decision-theoretic paradigm of maximizing expected utility. Currently, systems like Google Now and Siri mostly react to user's requests and commands using hand-crafted responses, but they cannot initiate intelligent communication and plan for longer term interactions. The reason is that they lack a clearly defined general objective of the interaction. Our main premise is that communication and interaction are types of action, so planning for communicative and interactive actions should be based on a unified framework of decision-theoretic planning. To facilitate this, the system's state of knowledge (a mental model) about the world has to include probabilistic representation of what is known, what is uncertain, and how things change as different events transpire. Further, the state of user's knowledge and intentions (the theory of the user's mind) needs to include precise specification of what the system knows, and how uncertain it is, about the user's mental model, and about her desires and intentions. The theories of mind may be further nested to form interactive beliefs. Finally, decision-theoretic planning proposes that desirability of possible sequences of interactive and communicative actions be assessed as expected utilities of alternative plans. We describe our preliminary implementation using the Open CYC system, called MARTHA, and illustrate it in action using two simple interactive scenarios.

1 Introduction

Apple's Siri and Google Now are both very useful personal assistants with access to reams of potentially useful data, but they lack the general ability to converse and interact – their responses are triggered mostly by user commands and requests. This paper lays out a methodology for creating intelligent interactive systems by incorporating ideas from cognitive science and decision theory (DT).

The notions of mental models and world models are firmly established in AI. Mental models are representations of an agent's beliefs, goals, and intentions. They can include facts about the environment describing, say, weather, traffic, prices at nearby sandwich shop, games and activities, etc. But, in order to interact effectively with other agents the mental models must also be able to keep track of the beliefs, goals, and intentions of other agents – this ability is called a theory of mind (TM) (Gallese and Goldman 1998; Frith and Frith 2005; Ohtsubo and Rapoport 2006). Indeed, for an agent to have a theory of mind, it must recognize that other agents act according to their own, usually unobservable, mental models. In particular, according to simulation theory of TM, an agent may think about how she might feel and think given environmental inputs received by another agent (Gallese and Goldman 1998; Shanton and Goldman 2010).

An intelligent personal assistant using a theory of mind must be able to track the user's mental model in terms of beliefs and desires, using knowledge to support the user in pursuit of his goals. Frequently, the assistant may find that the user may have incomplete or erroneous beliefs. For instance, the assistant may have access to databases which it knows the user does not have, so if the user believes that, say, the price of a sandwich nearby is $2 while it really is $5, the assistant may inform the user of the actual price. Note, however, that telling the user *everything* the assistant knows that the user does not know is recipe for disaster, so a reliable and consistent way of prioritizing the information is necessary. Likewise, telling the user something he already knows is (usually) useless, so the assistant should stop itself from being redundant.

In other cases it may be the assistant who knows less than the user. For example the system may sense that the user got into a car and is driving but the destination is unknown. We would like the assistant to be able to compute that asking user a question under these circumstances is the right thing to do. How? In our view, a theory of mind, including the information indicating preference, is *essential* for value-driven intelligent social behavior.

modes of interaction require deeply nested theories of mind. Consider the act of telling your friend Jim that you know John's phone number. Why did you find it useful to tell him this? The reason is that your model of Jim shows that he incorrectly believes that you do not know John's phone number – telling him corrects this, and now you both know the correct information. This is already a three-layer model. Going deeper, consider the act of telling Jim that you don't know John's phone number, but you know that Sally does.

Here, you've used the fact that you know that Jim thinks you know John's number – a three-layer model – concurrently with the fact that you know that Jim believes what you know about Sally is correct – a four-layer model!

It is known that humans can operate on four nested levels of modeling, but tend to lose track of information nested on deeper levels (Ohtsubo and Rapoport 2006). One can thus suppose that the skills in social interaction uses theories of mind nested at five or six levels at most.

The objective of our line of research is to design artificial agents that can match these capabilities. In order to do so, we need a general framework of processing information in nested theories of mind. We propose that a nested decision theoretic process should be used for this purpose. The key is to assign quantifiable values to an agent's desires and plans using utility functions. If anything about the world, or about other agents, is uncertain, the expected utility is the guide to optimal (and intelligent) ways to interact.

The central tenet to our approach is this: since communicative acts alter the other agent's mental state (which is reflected in the first agent's theory of mind), the optimal communicative act is the one which changes the theory of mind in the most beneficial way. Since actions (e.g., doing something) and mental states (e.g., believing something) can both have utility values, the change in utility can be determined by the total utility contributed by actions and states in a plan.

These plans should not necessarily be triggered by user prompts. It is possible to detach the planning process from user input so that plans are constantly being generated and evaluated with respect to the immediate state. Thus, if an act is useful at any time, it can and should be executed without a user request, just as humans do not always need to be prompted to volunteer information. Still, this does not preclude responding to a direct request for help as well.

In the remainder of this paper we detail an implementation of the ideas presented above. We used OpenCyc™[1] to apply the world model and theory of mind of a user in two simple scenarios. We call this implementation the **M**ental state–**A**ware **R**eal-time **TH**inking **A**ssistant, or **MARTHA**,[2] with the goal of creating a knowledge assistant capable of understanding of the user's information needs. We include an example run that results in Martha computing the optimal communicative act to be executed, given what is known. We also walk through a theoretical assistive search application. We conclude with possible avenues for future work.

2 Background & Related Work

There are two leading theories on the origin of theory of mind: theory theory and simulation theory. Theory theory (Gallese and Goldman 1998; Frith and Frith 2005) is the idea that humans acquire a theory of mind by associating mental states with observed behaviors and formulating commonsense theories of correlation. This is akin to how one gains an informal understanding of physical concepts, such as gravity, through observation. An example of this rule-based approach would be concluding a person is happy by observing him smile, having previously learned the correlation.

Intuitive evidence, however, favors simulation theory.[3] If Alice is trying to understand how Bob feels or thinks in a certain situation, she will likely "put herself in the Bob's shoes" by thinking about how she might feel, given the same environmental inputs as Bob. Simulation theory is exactly this intuitive process of simulating one's thought process in a hypothetical situation (Gallese and Goldman 1998; Shanton and Goldman 2010). The observer can perform an imaginary spatial translation into the point of view of the observed individual and determine a likely mental state attributable to the observed individual (Gallese and Goldman 1998; Frith and Frith 2005). Another proposal is that the observer can approximate the observed individual's mental state through a series of "inhibitions" on his own mental state (Leslie, Friedman, and German 2004)

Our implementation uses Cyc® – a project which aims to create a comprehensive general knowledge base to help intelligent agents extend to a broad range of applications (Matuszek et al. 2006; Ramachandran, Reagan, and Goolsbey 2005). Cyc is a structured representation of knowledge largely organized in first-order logical statements. It has a powerful and efficient inference engine that allows it to draw conclusions quickly with practical accuracy (Ramachandran, Reagan, and Goolsbey 2005). Interaction with the knowledge base proceeds through assertions and queries in CycL, a Lisp-like language created for Cyc. It is also accessible via a Java API. Our work uses OpenCyc, a small open-source portion of the proprietary Cyc database which the developers have released for general use.

Our implementation views planning, which acts on the above knowledge, as originating from connecting pre- and post-conditions of actions in pursuit of a goal. (Cantrell et al. 2012) not only successfully built a system capable of creating plans using known pre/post-conditions, but they also showed that the system could parse these conditions from verbal directions on-the-fly.

Previously, there have been attempts to implement rigorous assistive agents with mental modeling in the past. A notable example is PExA (Myers et al. 2007), a personal scheduling and work organization assistant for enterprise that was made to be integrated into the CALO (Tur et al. 2010) meeting assistant system. PExA was intended to free employees from rote tasks by learning how to do them from the user. For the tasks it could not do, PExA would check over the user's work to correct mistakes. Most interestingly, PExA was capable of proactively communicating with the user, reminding him about obligations and problems, due to its ability to monitor the user's mental state. We seek to build upon this ability with a focus of extending the mental modeling to multiple layers.

MARTHA aspires to combine ideas from each of these different lines of research. In order to make MARTHA an assistive AI, we must first create an intelligent agent with

[1] OpenCyc is a trademark and Cyc is a registered trademark of Cycorp, Inc.

[2] Also stylized as "Martha".

[3] This is not to say that theory theory is not useful, however; in building an intelligent computer system, it can be convenient to abstract many learned processes into discrete logical rules.

the ability to plan and act in real time, centered on a theory of mind.

3 Implementing Theories of Mind in OpenCyc

MARTHA is written in Java and the Cyc Java API. With these, Martha creates a theory of mind by nesting planning processes in layers of hypothetical contexts. These contexts correspond to the human cognitive activity of "putting oneself in another's shoes." Hence, these contexts are "sandboxed" or isolated so that assertions in them do not directly change the beliefs in the parent context. This allows Martha to attribute simulated thoughts to the user and act on them as such. The nested nature of planning is displayed in Figure 1.

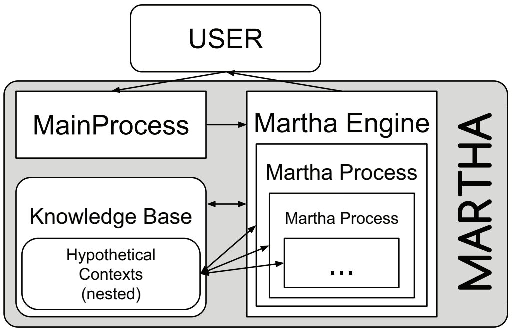


Figure 1: An organizational view of MARTHA. The arrows indicate the flow of information.

3.1 MARTHA's Modules

MARTHA is comprised of four primary modules.

The MainProcess module is responsible for initializing the knowledge base, spawning the Martha Engine, and finally accepting user input via a prompt line.

The knowledge base, implemented in OpenCyc, stores the entirety of Martha's knowledge about the world.

The Martha Engine module drives the real-time component of MARTHA by continuously interleaving **planning**, **evaluation**, and **execution phases**. The Martha Engine module initiates these cycles in the background, separate from the user prompt, so that Martha does not need to wait for user input before acting, allowing her to produce output of her own volition. Martha Engine also houses methods for evaluating the utility of actions and executing plans that interact with the user. It also contains a CycL interpreter. All operations on the OpenCyc knowledge base are directed through the Martha Engine so that it can keep track of the information it processes using meta-tags.

Martha's planning process is carried out by a series of nested Martha Processes spawned within the Martha Engine. The Martha Process contains algorithms for planning, as well as special evaluation and execution methods which operate across the nested structure. This planning takes place in the sandboxed hypothetical contexts containing propositions which could become true if some actions are executed. This is discussed in further depth in Section 3.4.

3.2 The OpenCyc Knowledge Base

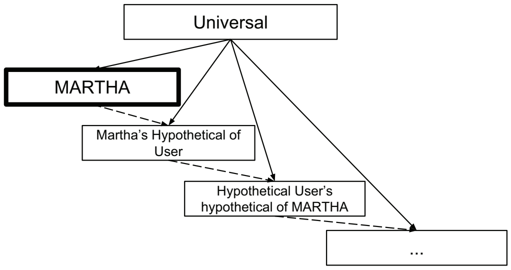

Universal
MARTHA
Martha's Hypothetical of User
Hypothetical User's hypothetical of MARTHA
...

Figure 2: The hierarchy of contextual spaces in MARTHA.

Martha's knowledge base (KB) is built on top of OpenCyc using the Java API. The KB is organized into the Universal context, the MARTHA context, and hypothetical contexts (Figure 2). All contexts inherit base assertions from the Universal context. When started, Martha moves into the MARTHA context, which contains run-time facts and conclusions (which are not necessarily universal). Hypothetical contexts inherit all universal facts, but *only* selected facts from their parent context (via the Hypothetical Context Constructor). Because each hypothetical context is isolated from its parent context, Martha is able to run simulations, i.e. perform assertions and observe results, without contaminating the parent and main MARTHA contexts.

The actual contents of the KB can be divided into the categories of facts, action definitions and pre/post-conditions, utility values, and miscellaneous rules.

Facts are assertions about constants and functions, such as `(isa Rover Dog)`. Goals, beliefs, and knowledge are three special kinds of facts. An agent's goals are represented with the `desires` predicate while beliefs and its subtype, knowledge, are represented with `beliefs` and `knows`.

More important to Martha are assertions about actions, especially their pre- and postconditions. These can be as simple as `(preconditionFor-Props (knows ?AGENT (basicPrice ?OBJECT ??VALUE)) (buys ?AGENT ?OBJECT))`, which states that a an agent must know the price of the object to buy it. But through the use of implications and conditional statements, these definitions can become quite complex: `(implies (and (beliefs ?AGENT (sells ?STORE ?PRODUCT)) (desires ?AGENT (buys ?AGENT ?PRODUCT))) (causes-PropProp (desires ?AGENT (buys ?AGENT ?PRODUCT)) (desires ?AGENT (at-UnderspecifiedLandmark ?AGENT ?STORE))))`. This says that given that an agent believes that a certain store sells a product which the agent wants to buy, the desire to buy a product will cause the agent to want to go to the store.

Equally as important and numerous are statements about the utility values of certain states and actions, which are placed in assertions like `(baseUtilityValue USER (driveTo USER ?PLACE) -10)`. This example states that the base utility value to the user of driving to a certain place is -10 (due to travel costs).

A key tool for organizing the knowledge base is the Hypothetical Context Constructor. This spawns nested sandboxed contexts for simulating layers in the theory of mind. Belief statements are unwrapped according to the ordering of the nested layers, using the nested belief statements of the current context to initialize the beliefs of the next context. For example, in a three layer simulation consisting of a Martha thought process, a user simulation, and a Martha simulation, the statement `(beliefs USER (beliefs MARTHA (isa Rover Dog))))` would be unwrapped to be `(beliefs MARTHA (isa Rover Dog)))` in the user simulation, and then `(isa Rover Dog)` in the simulation of the user simulating Martha. This makes it easy to package knowledge so that it can be injected directly into the knowledge base.

Finally, Martha also has a variety of Martha Functions which have little meaning within the OpenCyc KB but are indispensable to the Martha Engine. Some key functions are `baseUtilityValue`, `says`, `focus`, and `carryover`. `baseUtilityValue` specifies the unmodified utility value of a state to a particular agent as a parameter of a utility function. `says` is a functional predicate applied to statements which causes Martha to say those statements. `focus` is a meta-tag that inputs a fact, goal, or action as the seed of a forwards search. User statements are automatically wrapped in `focus` tags by the MainProcess. `carryover` is a meta-tag used by the Hypothetical Context Constructor to include the tagged fact in the next nested context. Carrying over a focus statement to see its implications is often very useful; thus there is also a `sowhat` function which is an alias for `(carryover (focus` *`statement`*`))`.

3.3 Shifting Focus

In intuitive conversation, individuals often discuss only a few topics at a time; it can be awkward to jump around, for instance, by first talking about politics and then about buying sandwiches, without precedent. Thus, it can be helpful to avoid extraneous lines of thought in MARTHA by using the `focus` predicate to center her planning on what is tagged.

Additionally, in real conversation, focuses shift rapidly. So, the `focus` is coupled with a "focus ticker," a counter to identify the latest set of focuses.[4] So, in order for a focus tag to be considered, it must have a number which corresponds to the focus ticker. Let us note that focuses are not the same as contexts; context here refers to assertion and inference contexts in the OpenCyc knowledge base.

3.4 Theories of Mind and Nested Planning

Simulation theory suggests that theory of mind arises when individuals extend their thought process into another individual's situation. In MARTHA, this is represented by applying Martha's planning (backward-search and forward-search) in a series of nested mental models. Each of these nested layers contains the beliefs of a simulated agent, created by the Hypothetical Context Constructor.

The planning phase begins when the Martha engine begins to "explore." It launches a forward-search planning process seeded with relevant `focus` statements. From these focuses, the search plans forwards in time, chaining preconditions of actions to postconditions. Concurrently, a backward-search occurs, starting with user goals and chaining in reverse. These run until a timeout or the search is exhausted. Each resulting chain of preconditions, actions, and postconditions is called a **plan**, and these are queued for evaluation. In the backward-search, unfulfilled preconditions become the focus of the planning phase in the next nested layer.

Martha is agnostic to which search scheme the plans originated from, since they are all series of viable actions and have independent, non-conflicting roles. The purpose of a forward-search is discovery; it is analogous to the question, "What if...?" which explores the consequences of actions. On the other hand, the purpose of the backward-search is to directly look for paths to user's goals (if known), seeking out unfulfilled preconditions in particular.

The evaluation portion of the planning phase (different from the evaluation *phase* in the Martha Engine) follows the search portion. Each plan is scored as the sum of the utility of its components. Plans must meet a minimum score to be considered; useless lines of search are discarded to maintain efficiency. In hypothetical contexts, these thresholds control how many eligible chains are passed on and picked up by the Hypothetical Context Constructor and injected into the next nested layer.

Once the planning phase reaches a maximum nesting depth, planning ends and the evaluation phase begins. Returning to the top layer, the Martha Engine scores all the proposed plans by their utility. Since plans in the Martha Engine may be executed into reality, they must meet a very high minimum score to be considered; only the best plan is executed – if it is even worth it! This threshold has a different role than the threshold in the evaluation portion of the planning phase in that it is designed to filter out plans with negligible utility which would cause Martha to "babble".

Martha Actions generated by the plans are briefly investigated as standalone actions to see where they might lead using forward-search. This is analogous to double-checking actions for hidden implications in actions. This is a key ability in social situations, as it can represent societal expectations for behavior.

After the evaluation phase is complete, the execution phase begins. If there is one, the single best plan that meets the threshold is read step by step in the Martha Engine. Steps that correspond to Martha Actions are executed in reality. Then, the cycle of Martha engine repeats, starting again at the planning phase.

The overall aim of the above implementation is to allow Martha to use simulations of the minds of other agents to identify their intentions and plans of action so that, as an assistive AI, it can act to help fulfill the inferred needs of these agents. With this recursive, nested planning simulation, Martha mimics an organic thought process characteristic to humans.

[4] One implication of this is that the counter increases regularly for each cycle of Martha engine. This produces a continually shifting focus and a notion of time.

4 Results

4.1 An Example Run with the Sandwich Scenario

The User is looking to buy a sandwich, specifically, the FiveDollarSteakSandwich (Figure 3). However, with a propensity to overlook the significance of names, he cannot tell if he can afford it. He knows that Martha knows the price of the sandwich, and so he says that he intends to buy a FiveDollarSteakSandwich, and that he has $4. From these two statements, Martha infers that the user would like to know whether he can afford the sandwich.

The programmatic setup for this scenario is created by a series of initial assertions in CycL (shown here in plain english):

1. Knowing that you can afford an item is a precondition to buying the item.
2. If you have less money than an item's price, then you cannot afford the item; if you have more than or the same as an item's price, then you can afford it.
3. You know that Martha will tell you whether you can afford something if she knows you want to know that.
4. If you try to buy something you can't afford, you will feel embarrassed.

With these facts in mind, the scenario and Martha's thought process are designed to work as follows:

Step 1. The user tells Martha that he wants to buy a FiveDollarSteakSandwich, and that he has $4.

Step 2. Martha considers the user input from Step 1 in the planning phase, asking itself why the user said what he said using the `sowhat` meta-tag.

Step 3. Martha thinks about what the user was thinking when he gave her the input. When he said "I have $4," and "I want to buy a FiveDollarSteakSandwich," he knew that would cause Martha to know those facts–creating nested beliefs which are fundamental to theory of mind. Martha also wonders about the user's desire to buy a FiveDollarSteakSandwich. She knows that he knows that to buy a product, one must first be able to afford it, so Martha reasons that the user must be wondering whether he can afford it.

Step 4. Martha simulates the user simulating Martha. Previously, Martha concluded that the user knows that Martha knows that he has $4 and that he wants to buy the sandwich. Given Initial Assertion 3, Martha knows that the user therefore expects her to tell him whether or not he can afford the sandwich. Notice how there is no rule governing *which* Martha should say, just an *expectation* that she will respond accordingly. This is because, realistically, the user cannot know for sure what Martha's internal rules are, but he can have social expectations for Martha's behavior. To see which is the most useful, both responses are queued for further investigation.

Step 5. Martha begins the evaluation phase to investigate these two plans. Note that the knowledge and conclusions from the planning phase are preserved in the MARTHA context. She also knows the sandwich costs $5.

Step 6. Martha explores the possibilities of a suggested action produced by the planning phase: telling the user he *can't* afford the sandwich. From Initial Assertion 1, Martha knows that if she says this, the user will know that he cannot afford the sandwich, and therefore cannot buy it because the mandatory precondition of being able to afford what one wants to buy is unfulfilled. Martha's speech act here is associated with a positive utility value because Martha is telling the user something he doesn't know.

Step 7. With a similar logic, Martha finds that if she tells the user that he *can* afford the sandwich, he will go ahead and try buying it, resulting in his embarrassment (since he can't afford it). This is associated with a strong negative utility value.

Step 8. Martha looks at the utility values of the proposed plans, and chooses the highest one which exceeds the minimum utility threshold.

Step 9. Martha executes the chosen plan, telling the user that he cannot afford the sandwich. The user is naturally disappointed, but glad he has been saved the embarrassment of trying to buy a sandwich he could not afford.

4.2 Selected Output from the Sandwich Scenario

We provide screenshots from the execution of the program to demonstrate MARTHA's capabilities. Note that MARTHA presently does not use natural language processing with OpenCyc, so communications are still performed through CycL assertions.

In Figure 4, we see the user interaction as described by the model above. The user tells Martha that he has $4 and wants the FiveDollarSteakSandwich, and she responds that the user cannot afford the sandwich. Interestingly, Martha also tells the user the price of the sandwich. This is a surprise: in a naïve planner, saying the price be part of a plan in which the user buys an item he cannot afford. While Martha initially avoided this plan in the first planning cycle, after she told the user that he couldn't afford the sandwich, she seemed to then consider *might* happen if the user *were* able to afford the sandwich. This latter speech act emerges as useful in the next cycle of the engine and is added moments later, reminiscent of a *second thought*.

In Figure 5, we looked at what might happen if the user changed his mind about which sandwich he wanted. The user begins by telling Martha that he has $4 and wants the FiveDollarSteakSandwich. Upon learning that he can't afford it, he tells Martha that he now wants the JimmyJohnny-BLT. As an additional challenge, he says to Martha that he already knows it costs $3.50. Martha correctly tells the user that he can afford it without saying the cost again, since he already knows.

These examples demonstrate how complex behavior – such as giving second-thoughts, thinking hypothetically, and correcting speech acts with new information – can arise from a set of common-sense facts and a nested planning algorithm. In this way, Martha can be adapted to a number of interactive settings, by integrating an appropriate knowledge base. We hope that by integrating a large and diverse amount of these, Martha can be extended to work in a broad range

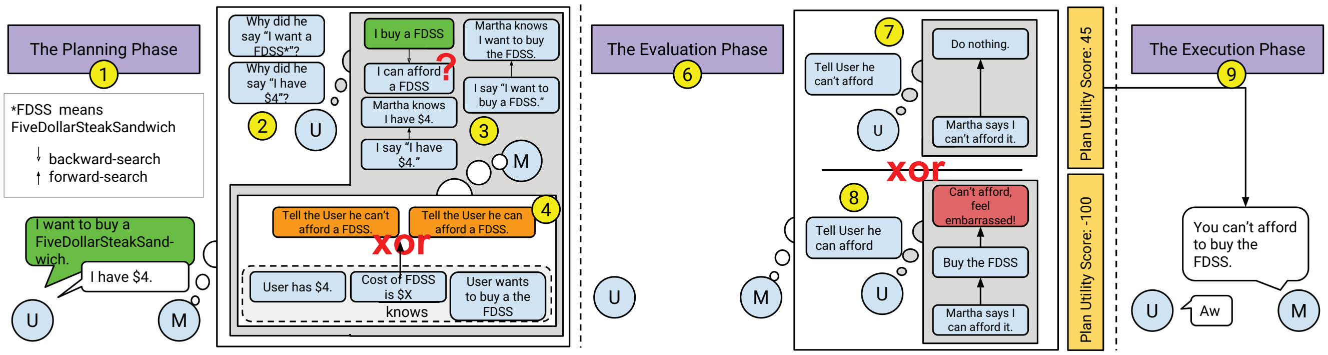
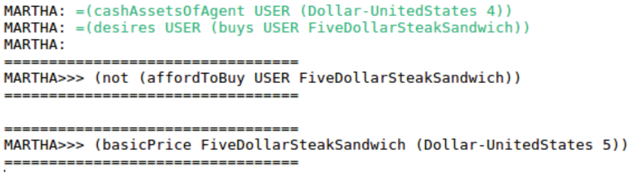

The Planning Phase
1
*FDSS means FiveDollarSteakSandwich
backward-search
forward-search
I want to buy a FiveDollarSteakSandwich.
I have $4.
U
M
Why did he say "I want a FDSS*"?
Why did he say "I have $4"?
2
U
I buy a FDSS
I can afford a FDSS
?
Martha knows I have $4.
I say "I have $4."
Martha knows I want to buy the FDSS.
I say "I want to buy a FDSS."
3
M
Tell the User he can't afford a FDSS.
Tell the User he can afford a FDSS.
4
xor
User has $4.
Cost of FDSS is $X
knows
User wants to buy a the FDSS
The Evaluation Phase
6
U
M
7
Tell User he can't afford
U
Do nothing.
Martha says I can't afford it.
xor
8
Tell User he can afford
U
Can't afford, feel embarrassed!
Buy the FDSS
Martha says I can afford it.
Plan Utility Score: 45
Plan Utility Score: -100
The Execution Phase
9
You can't afford to buy the FDSS.
U
Aw
M

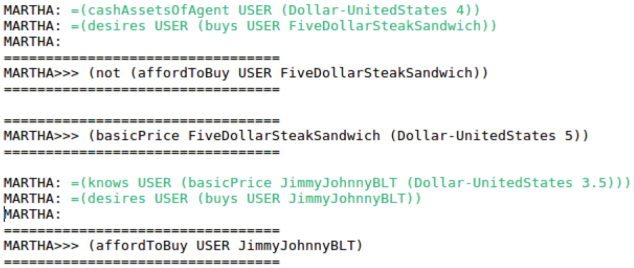
Figure 3: The thought process of the Sandwich Scenario. Provided only with information about how much money the User, U, has and which sandwich he wants, MARTHA, M, must infer that the user needs to know whether or not he can afford the sandwich before he goes to buy it. Through a series of nested steps, Martha is able to simulate the user's intentions, and Martha responds accordingly by telling the user whether he can afford the sandwich.



```
MARTHA: =(cashAssetsOfAgent USER (Dollar-UnitedStates 4))
MARTHA: =(desires USER (buys USER FiveDollarSteakSandwich))
MARTHA:
================================
MARTHA>>> (not (affordToBuy USER FiveDollarSteakSandwich))
================================

================================
MARTHA>>> (basicPrice FiveDollarSteakSandwich (Dollar-UnitedStates 5))
================================
```

Figure 4: The Sandwich Scenario output, as designed. Green text is user input, while black text is MARTHA output.



```
MARTHA: =(cashAssetsOfAgent USER (Dollar-UnitedStates 4))
MARTHA: =(desires USER (buys USER FiveDollarSteakSandwich))
MARTHA:
================================
MARTHA>>> (not (affordToBuy USER FiveDollarSteakSandwich))
================================

================================
MARTHA>>> (basicPrice FiveDollarSteakSandwich (Dollar-UnitedStates 5))
================================

MARTHA: =(knows USER (basicPrice JimmyJohnnyBLT (Dollar-UnitedStates 3.5)))
MARTHA: =(desires USER (buys USER JimmyJohnnyBLT))
MARTHA:
================================
MARTHA>>> (affordToBuy USER JimmyJohnnyBLT)
================================
```

Figure 5: MARTHA avoiding redundancy. She responds *without* telling the user the price of the sandwich again.

of activities.

5 Conclusions and Future Work

This paper puts forth what we consider to be principles of intelligent interaction and communication: decision-theoretic rationality and the use of mental models and nested theories of mind. We describe our implementation using OpenCyc through a sandwich purchase scenario.

The applications of MARTHA, of course, can be extended beyond mere sandwich shopping. Even the simple ability to tell the user whether or not he can afford something can be coupled with product data to allow Martha to aide users in financial decisions. By integrating the necessary knowledge with our foundational algorithm, Martha could be made to be capable of

- Providing information (like weather or traffic updates) when the user needs it by anticipating the user's intentions;
- Helping people, from families to investors, make sound financial decisions, using its nested planning algorithm;
- Assisting a child to find a book he wants to read, a researcher to find the perfect article, a government official to find a particular document, etc., by *understanding* what they are looking for through conversational feedback;
- Issuing dynamic reminders, such as a reminder to take a medication, when it is least likely to be ignored, rather than at an easily-dismissed pre-set time.

Most importantly, these individual behaviors can be implemented simultaneously in MARTHA. When outputs from one mode of operation can act as inputs to another because Martha has knowledge and function in those areas, it is evident that MARTHA can gain sophistication through an expansion of its knowledge base.

In future work, a number of issues need to be tackled to make our approach scale to reality. In addition to optimizing the algorithm for faster execution, these include keeping close track of the preferences and goals of the user (for example, by using inverse reinforcement learning) (Ng and Russell 2000); automatically inferring new rules representing regularities of every day life; handling overlapping goals and tasks; and keeping track of the user's current focus and attention span.

Ultimately, we see that the addition of a theory of mind to assistive AI has the potential to greatly improve human interaction with intelligent agents in that these can communicate more naturally and effectively. Agents capable of modeling mental states can not only avoid redundancy in communicative acts, but they can act more intelligently by predicting the motives and intentions of other agents. In MARTHA, we are confident that the system has the potential to bring contextual understanding to human conversations; with more work to enlarge its knowledge base and data acquisition capabilities as well as its algorithm, this could significantly advance assistive intelligence.

References

Cantrell, R.; Talamadupula, K.; Schermerhorn, P.; Benton, J.; Kambhampati, S.; and Scheutz, M. 2012. Tell me when and why to do it! Run-time planner model updates via natural language instruction. In *7th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 2012*, 471–478.

Frith, C., and Frith, U. 2005. Theory of mind. *Current Biology* 15(17):R644 – R645.

Gallese, V., and Goldman, A. 1998. Mirror neurons and the simulation theory of mind-reading. *Trends in Cognitive Sciences* 2(12):493 – 501.

Leslie, A. M.; Friedman, O.; and German, T. P. 2004. Core mechanisms in theory of mind. *Trends in Cognitive Sciences* 8(12):528 – 533.

Matuszek, C.; Cabral, J.; Witbrock, M.; and Deoliveira, J. 2006. An introduction to the syntax and content of Cyc. In *Proceedings of the 2006 AAAI Spring Symposia*, 44–49.

Myers, K.; Berry, P.; Blythe, J.; Conley, K.; Gervasio, M.; McGuinness, D. L.; Morley, D.; Pfeffer, A.; Pollack, M.; and Tambe, M. 2007. An Intelligent Personal Assistant for Task and Time Management. *AI Magazine* 28(2):47 – 61.

Ng, A. Y., and Russell, S. J. 2000. Algorithms for Inverse Reinforcement Learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, 663–670. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Ohtsubo, Y., and Rapoport, A. 2006. Depth of reasoning in strategic form games. *The Journal of Socio-Economics* 35(1):31 – 47. Essays on Behavioral Economics.

Ramachandran, D.; Reagan, P.; and Goolsbey, K. 2005. First-orderized researchcyc: Expressivity and efficiency in a common-sense ontology. In *AAAI Workshop on Contexts and Ontologies: Theory, Practice and Applications*.

Shanton, K., and Goldman, A. 2010. Simulation theory. *Wiley Interdisciplinary Reviews: Cognitive Science* 1(4):527–538.

Tur, G.; Stolcke, A.; Voss, L.; Peters, S.; Hakkani-Tur, D.; Dowding, J.; Favre, B.; Fernandez, R.; Frampton, M.; Frandsen, M.; Frederickson, C.; Graciarena, M.; Kintzing, D.; Leveque, K.; Mason, S.; Niekrasz, J.; Purver, M.; Riedhammer, K.; Shriberg, E.; Tien, J.; Vergyri, D.; and Yang, F. 2010. The CALO Meeting Assistant System. *Audio, Speech, and Language Processing, IEEE Transactions on* 18(6):1601–1611.

Toward Beneficial Human-Level AI… and Beyond

Philip C. Jackson, Jr.
TalaMind LLC
www.talamind.com
dr.phil.jackson@talamind.com

Abstract

This paper considers ethical, philosophical, and technical topics related to achieving beneficial human-level AI and superintelligence. Human-level AI need not be human-identical: The concept of self-preservation could be quite different for a human-level AI, and an AI system could be willing to sacrifice itself to save human life. Artificial consciousness need not be equivalent to human consciousness, and there need not be an ethical problem in switching off a purely symbolic artificial consciousness. The possibility of achieving superintelligence is discussed, including potential for 'conceptual gulfs' with humans, which may be bridged. Completeness conjectures are given for the 'TalaMind' approach to emulate human intelligence, and for the ability of human intelligence to understand the universe. The possibility and nature of strong vs. weak superintelligence are discussed. Two paths to superintelligence are described: The first path could be catastrophically harmful to humanity and life in general, perhaps leading to extinction events. The second path should improve our ability to achieve beneficial superintelligence. Human-level AI and superintelligence may be necessary for the survival and prosperity of humanity.

The Future of Humanity: Ethics and AI

Some potential consequences of general artificial intelligence were outlined in (Jackson 1974). Two possibilities for the "harvest of AI" were briefly discussed: A world with the machine as dictator, and a world with "well-natured machines" having enormous benefits to humanity.

Relatively recent work on 'artificial general intelligence' (Goertzel and Pennachin 2007) has included substantive research on AGI's potential consequences for humanity: Bostrom, Omohundro, Tegmark, Yudkowsky and others have discussed future scenarios in which AGI could lead to superintelligent systems with good or bad conduct toward humanity. AGI may be necessary for the survival and prosperity of humanity but if AGI is not developed very carefully it could lead to the extinction of humanity.

Ethics is the branch of philosophy that studies concepts of right and wrong (good and bad) conduct. Until recently ethics has only needed to focus on conduct by humans. Ethics and AI research now intersect regarding concepts of right and wrong conduct by intelligent machines, and right and wrong conduct in human applications of intelligent machines.

This is a challenge for AI scientists because ethical concepts of right and wrong go beyond simple questions of whether factual or theoretical knowledge is true or false, or whether problem solving behavior is successful or unsuccessful. In general, we cannot expect concepts of right and wrong conduct to be easily understood by machines. It can be a challenge for humans to distinguish these concepts sometimes.

Yet if the survival and prosperity of humanity are at stake, we are obligated to accept the challenge. Hence this paper will consider ethical, philosophical, and technical topics related to achieving beneficial human-level AI and superintelligence. The term 'beneficial' in this context does not seem to have any rigorous, agreed-upon definition. It will be used broadly to refer to consequences that are positive for humanity and biological life[1] in general.

The Possibility of Human-Level AI

A first question is whether human-level AI is even possible: The 'TalaMind thesis' (Jackson 2014) presents a research approach toward human-level artificial intelligence, which will support this paper's discussion of human-level AI's implications for the future of humanity.

The thesis endeavors to address all the major theoretical issues and objections that might be raised against its approach, or against the possibility of achieving human-level AI in principle. No insurmountable objections are identified, and arguments refuting several objections are presented. Thesis section 7.8 gives reasons in favor of the TalaMind approach over other approaches to human-level AI.

[1] Life based on DNA that has been developed by evolution. (Cf. Tegmark 2017).

The approach involves developing an AI system using a language of thought (called 'Tala') based on the unconstrained syntax of a natural language; designing the system as an 'intelligence kernel', i.e. a collection of concepts that can create and modify concepts, expressed in the language of thought, to behave intelligently in an environment; and using methods from cognitive linguistics such as mental spaces and conceptual blends for multiple levels of mental representation and computation.

Proposing a design inspection alternative to the Turing Test, the thesis discusses 'higher-level mentalities' of human intelligence, which include natural language understanding, higher-level learning, meta-cognition, imagination, and artificial consciousness.

'Higher-level learning' refers collectively to forms of learning required for human-level intelligence such as learning by creating explanations and testing predictions about new domains based on analogies and metaphors with previously known domains, reasoning about ways to debug and improve behaviors and methods, learning and invention of natural languages and language games, learning or inventing new representations, and in general, self-development of new ways of thinking. The phrase 'higher-level learning' is used to distinguish these from previous research on machine learning. (Cf. Valiant 2013; Goertzel and Monroe 2017)

The thesis discusses an architecture called TalaMind for design of systems following its approach. The architecture is open, e.g. permitting predicate calculus and conceptual graphs in addition to the Tala language, and permitting deep neural nets and other methods for machine learning.

The thesis describes the design of a prototype demonstration system, and discusses processing in the system that illustrates the potential of the research approach to achieve human-level AI.

Of course, the thesis does not claim to actually achieve human-level AI. It only presents a theoretical direction that may eventually reach this goal, and identifies areas for future AI research to further develop the approach. These include areas previously studied by others which were outside the scope of the thesis, such as ontology, common sense knowledge, spatial reasoning and visualization, etc.

The TalaMind approach is similar though not identical to the 'deliberative general intelligence' approach proposed by (Yudkowsky 2007), as discussed in (Jackson 2014, §2.3.3.5). The architectural diagrams for human-like general intelligence given by (Goertzel, Iklé, and Wigmore 2012) may be considered as design aspects for TalaMind.

Human-Level AI ≠ Human-Identical AI

The TalaMind thesis gives reasons why the Turing Test does not serve as a good definition of the goal we are trying to achieve, human-level AI. In particular, the Turing Test conflates human-level intelligence with human-identical intelligence, i.e. intelligence indistinguishable from humans. This is important because in seeking to achieve human-level AI we need not seek to replicate human thinking. Human-level AI can be 'human-like' without being human-identical. (Jackson 2014, §2.1.1)

In particular for beneficial AI, the concept of self-preservation could be quite different for a human-level AI than it is for a human. A human-level AI could periodically backup its memory, and if it were physically destroyed, it could be reconstructed and its memory restored to the backup point. So even if it had a goal for self-preservation, a human-level AI might not give that goal the same importance a human being does. It might be more concerned about protection of the technical infrastructure for the backup system, which might include the cloud, and by extension, civilization in general.

A human-level AI could understand that humans cannot backup and restore their minds, and regenerate their bodies if they die, at least with present technologies. It could understand that self-preservation is more important for humans, than for AI systems. The AI system could be willing to sacrifice itself to save human life, especially knowing that as an artificial system it could be restored.

Artificial Consciousness

The TalaMind thesis accepts the objection by some AI skeptics that a system which is not aware of what it is doing, and does not have some awareness of itself cannot be considered to have human-level intelligence. The perspective of the thesis is that it is both necessary and possible for a system to demonstrate at least some aspects of consciousness, to achieve human-level AI. However, the thesis does not claim AI systems will achieve the subjective experience humans have of consciousness.

The thesis adapts the "axioms of being conscious" proposed by Aleksander and Morton (2007) for research on artificial consciousness. To claim a system achieves artificial consciousness it should demonstrate:

Observation of an external environment.

Observation of itself in relation to the external environment.

Observation of internal thoughts.

Observation of time: of the present, the past, and potential futures.

Observation of hypothetical or imaginative thoughts.

Reflective observation: Observation of having observations.

To observe these things, a TalaMind system should support representations of them, and support processing such representations. The TalaMind prototype illustrates how a TalaMind architecture could support artificial consciousness.

Symbolic Artificial Consciousness ≠ Human Consciousness

The axioms of artificial consciousness can be implemented with symbolic representations and symbolic processing, as illustrated in the TalaMind prototype. The human first-person subjective experience of consciousness is much richer and more complex. Achieving human-level AI may not require achieving human-identical consciousness in an AI system.

This is important to note because some authors seem to assume artificial consciousness will be equivalent to human consciousness, and assume a system with artificial consciousness should automatically have the same moral status and legal protections as a human being, so that switching off the system could be immoral or illegal. Some even suggest that if a system simulates consciousness within itself, and then terminates the simulation, the system may have performed a 'mind crime'. (Bostrom 2014)

Such suggestions are at best philosophical, and at worst Orwellian, if a system with symbolic artificial consciousness does not have any subjective experiences approaching human consciousness. Switching off such a system is not worse than switching off any computer that performs symbolic processing. Whether it is right or wrong to stop such a system depends on whether its symbolic processing would cause actions that affect human lives and biological life in general. This may be a simple or complex ethical decision, depending on whether the actions would be harmful or beneficial, or neither, or a combination of both.

Further, to support reasoning about potential future events, and counterfactual reasoning about past and present events, a system may need to simulate what other intelligent systems and people may think or do, and then terminate its simulations. The TalaMind thesis (Jackson 2014) uses the term 'nested conceptual simulation' to refer to an agent's conceptual processing of hypothetical scenarios, with possible branching of scenarios based on alternative events, such as choices of simulated agents. This amounts to a Theory of Mind capability within a TalaMind architecture, i.e. the ability of an AI system to consider itself and other systems or people as having minds with beliefs, goals, etc. Such simulations will be necessary for human-level AI, and should not be considered mind crimes.

For the same reason, relying on robots with such limited, symbolic artificial consciousness is not a form of 'slavery'. It is just symbolic processing.

The Possibility of Superintelligence

Since one of the abilities of human intelligence is the ability to design and improve machines, it's natural to suppose human-level AI could be applied to improve itself, and to think this might lead to "runaway" increases in machine intelligence beyond the human level. This possibility was first suggested[2] by Good (1965), and later considered by Vinge (1993), Moravec (1998), Kurzweil (2005), and others. Bostrom (2014) and Tegmark (2017) give current discussions.

To evaluate whether superintelligence can be achieved, let's consider what it could mean to "improve" human-level artificial intelligence, and whether and how human-level AI could improve itself to achieve superintelligence.

Here's a list of ways human-level AI could be improved relative to human intelligence:

Sensory capabilities – An AI system could perceive light (and sound) at different wavelengths, and phenomena at different scales (smaller or larger) than humans can directly observe.

Active capabilities – An AI system could perform actions at different physical scales than humans can directly perform.

Speed of thought – A computer can perform logical operations at speeds orders of magnitude faster than a neuron can fire. This may translate to corresponding speedups in thought.

Information access – An AI system could in principle access all the information in Wikipedia, or even the entire Web. A human-level AI could understand much of this information.

Extent and duration of memory – An AI system could in principle remember everything it has ever observed. Only a few humans claim this ability.

Duration of thought – A human-level AI could continue thinking about a particular topic for years, decades, centuries, millennia, … .

Community of thought – A collection of human-level AI's could share thoughts (conceptual structures) more directly, more rapidly, and less ambiguously than a collection of humans. If human-level AI can be copied and processed inexpensively, then much larger groups of

[2] Two earlier related suggestions are noteworthy: Turing (1950) asked "Can a machine be made to be super-critical?" i.e. to generate ideas in a manner analogous to super-criticality of nuclear reactions. Ulam (1958) recalled a conversation with von Neumann "on the ever accelerating progress of technology…which gives the appearance of approaching some essential singularity in the history of the race beyond which human affairs, as we know them, could not continue."

human-level AI's could be assembled to collaborate on a topic than would be possible with humans.[3]

Nature of thought – A human-level AI (or community of HLAI's) can develop new concepts and new conceptual processes. Such concepts and processes may be developed more rapidly than humans develop or understand them, creating *'conceptual gulfs'* in understanding between AI systems and humans.

Recursive self-improvement – This term does not seem to have any rigorous, agreed-upon definition though it is frequently used to describe how superintelligence could be achieved. Essentially it could be the recursive compounding of all the above improvement methods, and any other specific methods which may be identified.

These characteristics might all be described as 'more and faster' human-level AI, and may be called 'weak' superintelligence (cf. Vinge 1993). If human-level AI is achieved then it will be possible to create weak superintelligence.

Completeness of the TalaMind Approach

In effect, the TalaMind thesis (Jackson 2014) conjectures that the extensible 'nature of thought' for a TalaMind architecture is complete for supporting human-level AI, since it includes concepts represented in natural language as well as mathematically and logically in formal languages, supported by conceptual levels for cognitive concept structures and associative processing, with future extensions for spatial reasoning and visualization, etc. The Tala language is also a simple universal programming language for representing executable concepts and conceptual processes. In principle, TalaMind architectures could be extended to include human-level subjective consciousness, though that is a topic for a separate, future paper. This paper focuses only on the potential for AI systems with symbolic artificial consciousness, as discussed above.

The nature of thought for human intelligence is very powerful and extensible: It has enabled Homo sapiens to transition from "an unexceptional savannah-dwelling primate to become the dominant force on the planet" (Harari 2015). This transition has leveraged the expressive power and extensibility of human natural languages, which have enabled Sapiens to represent and communicate thoughts in domains of objective knowledge about the world such as physics and biology, and intersubjective knowledge about concepts invented by humans, such as money, corporations, ethical concepts, laws, nations, etc.

Although humans have cognitive biases and individual limitations, it may not be hubris to conjecture human intelligence is completely general. Consider that scientists and mathematicians have extended human concepts into new domains not directly observed, conceptualizing multiple dimensions, universal computation, general relativity, quantum theory, etc. If human intelligence is completely general then humans may eventually understand all the phenomena in the universe, by combining abilities to invent and represent hypothetical concepts about the universe with abilities to scientifically test hypotheses – if all the phenomena in the universe can be explained by practically testable theories. That's a big "if" of course.

If the TalaMind approach can achieve human-level AI, then a completeness conjecture for human intelligence extends to the TalaMind approach, and to superintelligent systems using TalaMind architectures.

Getting Over Conceptual Gulfs

Conceptual gulfs happen normally between human minds: For example, scientists have developed concepts that are not understood by the average person, or even by scientists in other fields. The worldwide scientific community may be considered superintelligent relative to any individual human. People accept this form of superintelligence because they believe scientific ideas can be understood and validated between scientists, and they believe scientific knowledge in general is beneficial to humanity.

Likewise, conceptual gulfs between weak superintelligence and humans could be bridged and new concepts could be explained to humans. This will be facilitated if AI systems follow the TalaMind approach, using a language of thought based on a natural language. Conceivably, conceptual gulfs between weak superintelligence and humans may have short duration in some domains, though there may always be conceptual gulfs to bridge.

Is 'Strong' Superintelligence Possible?

Could a strong superintelligence exist, qualitatively superior to weak superintelligence, i.e. superior to 'more and faster' human-level AI?

The answer seems to depend on other limits and characteristics of human intelligence that are not yet known by scientists. For instance, it appears not yet known for certain whether human intelligence requires super-Turing computation or quantum computation. Even if Penrose and Hameroff's "Orch-OR" hypothesis is disproved, the possibility may remain that other forms of nanoscale quantum computation occur within the brain. Neuroscientists may

[3] The TalaMind hypotheses do not require a society of mind architecture, but it is natural to implement a society of mind at the linguistic level of a TalaMind architecture. A society of mind architecture could also support a community of thought for human-level AI's. (Cf. Jackson 2014, §2.3.3.2.1)

consider this unlikely, but so far as I know it has not been completely ruled out. The same situation may hold for super-Turing computation.

If these forms of computation are required by the brain to support human intelligence, then human-level AI would need to include them to match the abilities of human intelligence. If human intelligence is also completely general, then no stronger form of intelligence would exist other than 'more and faster' human-level intelligence, i.e. weak superintelligence.

On the other hand, if these forms of computation are not used by the brain then extending human-level AI to use them could yield a 'strong' superintelligence, able to solve some problems that would be intractable for 'more and faster' human-level intelligence. Likewise, if human intelligence is not completely general then making human-level AI completely general could yield a strong superintelligence surpassing 'more and faster' human-level intelligence.

In either case, conceptual gulfs between humans and strong superintelligence could be bridged at least to the extent of using natural language to give descriptions of concepts developed by strong superintelligence.

Two Paths to Superintelligence

There are at least two somewhat different paths toward superintelligence. One path would focus on recursive self-improvement of general AI systems (AGI) having unchangeable 'final goals' which may be relatively simple and arbitrary. Bostrom (2014) discussed several ways this path could achieve superintelligence that would be catastrophically harmful to humanity and life in general, perhaps leading to extinction events.

Yudkowsky (2008) noted the design space for AGI is much larger than human intelligence, writing "The term 'Artificial Intelligence' refers to a vastly greater space of possibilities than does the term 'Homo sapiens.'" He strongly urged readers not to assume a fully general optimization process for AGI will be beneficial to humanity, yet advised not writing off the challenge of beneficial AI.

A second path toward superintelligence, consistent with the TalaMind approach, focuses on limiting the research design space to AI systems which have generality and which also have higher-level mentalities that are characteristic of human intelligence. This design space would be further limited to systems for which the only unchangeable goals are ethical goals beneficial to humanity and to biological life in general. This narrowing of the design space should improve our ability to achieve beneficial human-level AI and beneficial superintelligence via recursive self-improvement.

Human-Level Intelligence & Goals

In discussing the first path to superintelligence, Bostrom[4] (2014) relied on an 'orthogonality thesis' that "intelligence and final goals are independent variables: any level of intelligence could be combined with any final goal." He wrote:

> "There is nothing paradoxical about an AI whose sole final goal is to count the grains of sand on Boracay, or to calculate the decimal expansion of pi, or to maximize the total number of paperclips that will exist in its future light cone. In fact, it would be easier to create an AI with simple goals like these than to build one that had a human-like set of values and dispositions."

In taking the second path to superintelligence, these would not be allowed as unchangeable final goals. A TalaMind system would realize it is pointless to count the grains of sand on Boracay, impossible to fully calculate the infinite decimal expansion of pi, and harmful to humanity to maximize the number of paperclips in its future light cone. So it would reject or abandon these simple goals.

Bostrom (2014) also relied on an 'instrumental convergence thesis' that "superintelligent agents having any of a wide range of final goals will nevertheless pursue similar intermediary goals because they have common instrumental reasons to do so." In particular, he cited two instrumental goals which could cause superintelligent systems to be very harmful to humanity, perhaps leading to an extinction event. The first is a goal of self-preservation. The second is a goal of maximizing available resources. I've described above how a human-level AI could have a different concept of self-preservation, facilitating self-sacrifice to save human life. This could apply also to a superintelligence.

In scenarios (Bostrom 2014) discussed, the goal of maximizing resources causes a superintelligent system to accumulate as much money and power as possible, leading to very harmful consequences for humanity. This is another case where the ability to think ethically about goals, and change or abandon them is important. A human-level AI should understand there are appropriate and inappropriate relationships between goals and possible means to achieve goals. It should understand that achieving an important goal does not justify acquiring as much money and power as possible – rather, it should have an ethical meta-goal to achieve its goals with as little resources and money as possible, and without acquiring power over human lives or human decisions.

Taking the second path won't be easier than the first path just because the design space is smaller. Framing ethical goals and creating human-level AI systems which distinguish right from wrong conduct will be very difficult,

[4] Bostrom (2014) consolidated research on the first path by himself and others, including Omohundro and Yudkowsky.

but it needs to be done. TalaMind's use of a natural language mentalese will facilitate representing ethical concepts and goals.

To achieve beneficial AI it's also important to develop the TalaMind approach because a system that reasons in a conceptual language based on English (or some other common natural language) will be more open to human inspection than a black box or a system with an internal language that's difficult for people to understand.

Looking Forward

Human-level AI and superintelligence could help develop scientific knowledge more rapidly than possible through human thought alone, and help advance medicine, agriculture, energy systems, environmental sciences, and other areas of knowledge directly benefitting human prosperity and survival.

Human-level AI may be necessary for the long-term survival and prosperity of humanity: People are not biologically suited for lengthy space travel with present technologies. To avoid depleting the Earth's resources and to avoid the fate of the dinosaurs (whether from asteroids or super-volcanoes) our species will need economical, self-sustaining settlements off the Earth. Human-level AI may be necessary for mankind to spread throughout the solar system, and later the stars.

What Turing wrote in 1950 is still true: "We can only see a short distance ahead, but we can see plenty there that needs to be done." We have travelled far over six decades, and can now see a path toward beneficial superintelligence.

References

Aleksander, I. and Morton, H. 2007. Depictive architectures for synthetic phenomenology. In *Artificial Consciousness*, 67-81, ed. Chella, A. and Manzotti, R. Imprint Academic.

Bostrom, N. 2014. *Superintelligence – Paths, Dangers, Strategies*. Oxford University Press.

Bello, P. and Bringsjord, S. 2013. On how to build a moral machine. *Topoi*, 32, 2, 1-25.

Bringsjord, S., Arkoudas, K. and Bello, P. 2006. Toward a general logicist methodology for engineering ethically correct robots. *IEEE Intelligent Systems*, July 2006, 38-44.

Doyle, J. 1983. A Society of Mind – multiple perspectives, reasoned assumptions, and virtual copies. *Proceedings 1983 International Joint Conference on Artificial Intelligence*, 309-314.

Fauconnier, G. and Turner, M. 2002. *The Way We Think – Conceptual Blending and the Mind's Hidden Complexities.* New York: Basic Books.

Goertzel, B. and Pennachin, C. eds. 2007. *Artificial General Intelligence*. Springer.

Goertzel, B., Iklé, M. and Wigmore, J. 2012. The architecture of human-like general intelligence. *Foundations of Artificial General Intelligence*, 1-20.

Goertzel, B. and Monroe, E. 2017. Toward a general model of human-like general intelligence. *AAAI Fall Symposium Series Technical Reports*, FSS-17-05, 344-347.

Good, I. J. 1965. Speculations concerning the first ultraintelligent machine. *Advances in Computers*, vol. 6, 1965.

Hameroff, S. and Penrose, R. 2014. Consciousness in the universe: A review of the 'Orch OR' theory. *Physics of Life Reviews*, 11, 1, 39 – 78. Elsevier.

Harari, Y. N. 2015. *Sapiens: A Brief History of Humankind.* HarperCollins Publishers.

Jackson, P. C. 1974. *Introduction to Artificial Intelligence*. New York: Mason-Charter Publishers.

Jackson, P. C. 1985. *Introduction to Artificial Intelligence*, Second Edition. New York: Dover Publications.

Jackson, P. C. 2014. Toward Human-Level Artificial Intelligence – Representation and Computation of Meaning in Natural Language. Ph.D. Thesis, Tilburg University, The Netherlands.

Jackson, P. C. 2017. Toward human-level models of minds. *AAAI Fall Symposium Series Technical Reports*, FSS-17-05, 371-375.

Kurzweil, R. 2005. *The Singularity Is Near: When Humans Transcend Biology*. Viking.

Moravec, H. P. 1998. *Robot: Mere Machine to Transcendent Mind.* Oxford University Press.

Omohundro, S. M. 2008. The basic AI drives. In *Artificial General Intelligence 2008: Proceedings of the First AGI Conference*, ed. P. Wang, B. Goertzel & S. Franklin, 483-492.

Scheutz, M. 2017. The case for explicit ethical agents. *AI Magazine*, 38, 4, 57-64.

Tegmark, M. 2017. *Life 3.0: Being Human in the Age of Artificial Intelligence*. Alfred A. Knopf.

Turing, A. M. 1950. Computing machinery and intelligence. *Mind*, 59, 433 - 460.

Ulam, S. 1958. Tribute to John von Neumann, *Bulletin of the American Mathematical Society*, 64, 3, 1 - 49.

Valiant, L. G. 2013. *Probably Approximately Correct – Nature's Algorithms for Learning and Prospering in a Complex World.* Basic Books.

Vinge, V. 1993. The coming technological singularity: how to survive in the post-human era. *Whole Earth Review*, Winter 1993.

Walsh, T. 2017. The singularity may never be near. *AI Magazine*, 38, 3, 58 - 62.

Wilks, Y. 2017. Will there be superintelligence and would it hate us? *AI Magazine*, 38, 4, 65-70.

Yudkowsky, E. 2007. Levels of organization in general intelligence. In *Artificial General Intelligence*, ed. B. Goertzel & C. Pennachin, 389-501.

Yudkowsky, E. 2008. Artificial intelligence as a positive and negative factor in global risk. In *Global Catastrophic Risks*, ed. N. Bostrom & M. M. Ćircović, 308-345. Oxford University Press.

Preferences and Ethical Principles in Decision Making

Andrea Loreggia
University of Padova
andrea.loreggia@gmail.com

Nicholas Mattei
IBM Research
n.mattei@ibm.com

Francesca Rossi
IBM Research
University of Padova
francesca.rossi2@ibm.com

K. Brent Venable
Tulane University
kvenabl@tulane.edu

Abstract

If we want AI systems to make decisions, or to support humans in making them, we need to make sure they are aware of the ethical principles that are involved in such decisions, so they can guide towards decisions that are conform to the ethical principles. Complex decisions that we make on a daily basis are based on our own subjective preferences over the possible options. In this respect, the CP-net formalism is a convenient and expressive way to model preferences over decisions with multiple features. However, often the subjective preferences of the decision makers may need to be checked against exogenous priorities such as those provided by ethical principles, feasibility constraints, or safety regulations. Hence, it is essential to have principled ways to evaluate if preferences are compatible with such priorities. To do this, we describe also such priorities via CP-nets and we define a notion of distance between the ordering induced by two CP-nets. We also provide tractable approximation algorithms for computing the distance and we define a procedure that uses the distance to check if the preferences are *close enough* to the ethical principles. We then provide an experimental evaluation showing that the quality of the decision with respect to the subjective preferences does not significantly degrade when conforming to the ethical principles.

Introduction

If we want people to trust AI systems, we need to provide them with the ability to discriminate between good and bad decisions. The quality of a decision should not be based only on the preferences or optimisation criteria of the decision makers, but also on other properties related to the impact of the decision, such as whether it is ethical, or if it complies to constraints and priorities given by feasibility constraints or safety regulations.

A lot of work has been done to understand how to model and reason with subjective preferences. This is understandable, since preferences are ubiquitous in everyday life. We use our own subjective preferences whenever we want to make a decision to choose our most preferred alternative. Therefore the study of preferences in computer science and AI has been very active for a number of years with important theoretical and practical results (Domshlak *et al.* 2011; Pigozzi *et al.* 2015) as well as libraries and datasets (Mattei and Walsh 2013).

Our preferences may apply to one or more of the individual components, rather than to an entire decision. For example, if we need to choose a car, we may prefer certain colours over others, and we may prefer certain brands over others. We may also have conditional preferences, such as in preferring red cars if the car is a convertible. For these scenarios, the CP-net formalism (Boutilier *et al.* 2004) is a convenient and expressive way to model preferences (Rossi *et al.* 2011; Chevaleyre *et al.* 2008; Goldsmith *et al.* 2008; Cornelio *et al.* 2013) CP-nets indeed provide an effective compact way to qualitatively model preferences over outcomes (that is, decisions) with a combinatorial structure. CP-nets are also easy to elicit and provide efficient optimization reasoning (Chevaleyre *et al.* 2011; Allen *et al.* 2015). Moreover, in a collective decision making scenario, several CP-nets can be aggregated, e.g., using voting rules (Conitzer *et al.* 2011; Mattei *et al.* 2013; Cornelio *et al.* 2015), to find compromises and reach consensus among decision makers.

If ethical constraints are added to this scenario, it means that the subjective preferences of the decision makers is not the only source of information we should consider (Sen 1974; Thomson 1985; Bonnefon *et al.* 2016). Indeed, depending on the context, we may have to consider specific ethical principles derived from an appropriate ethical theory (Copp 2005). While preferences are important, when preferences and ethical principles are in conflict, the principles should override the subjective preferences of the decision maker. For example, in a hiring scenario, the preferences of the hiring committee members over the candidates should be measured against ethical guidelines and laws e.g., ensuring gender and minority diversity. Therefore, it is essential to have principled ways to evaluate if preferences are compatible with a set of ethical principles, and to measure how much these preferences deviate from the ethical principles. The ability to precisely quantify the distance between subjective preferences and external priorities, such as those given by ethical principles, provides a way to both recognize deviations from feasibility or ethical constraints, and also to suggest more compliant decisions.

In this paper we use CP-nets to model both exogenous priorities, e.g., those provided by ethical principles, and subjective preferences of decision makers. Thus the distance between an individual subjective preferences and some ethical principles can be measured via a notion of distance between CP-nets. Indeed, we define such a notion of distance (formally a distance function or metric) between CP-nets. A more comprehensive discussion of CP-nets and distances between them is given by Loreggia *et al.* (2018).

Since CP-nets are a compact representation of a partial order over the possible decisions, the ideal notion of distance is a distance between the induced partial orders of the CP-nets. However, the size of the induced orders is exponential in the size of the CP-net, and we conjecture that computing a distance between such partial orders is computationally intractable because of this possibly exponential explosion. Therefore we propose a tractable approximation that is computed directly over the CP-nets dependency graphs, and we study the quality of the approximation.

To define the desired distance between partial orders, we generalize the classic (Kendall 1938) τ (KT) distance, which counts the number of inverted pairs between two complete, strict linear orders. We add a penalty parameter p defined for partial rankings as proposed by (Fagin *et al.* 2006), and use this distance, that we call KTD, to compare partial orders. In KTD the contribution of pairs of outcomes that are ordered in opposite ways is 1 and that of those that are ordered in one partial order and incomparable in the other is p. We show that $0.5 \leq p < 1$ is required for KTD to be a distance.

For the tractable approximation of KTD, we can define a distance between CP-nets, called CPD, that only analyzes the dependency structure of the CP-nets and their CP-tables. We then characterize the case when $CPD = 0$, which correspond to when the two CP-nets have the same dependency structure and CP-tables. In other words, $CPD = 0$ if and only if the two CP-nets are identical and they induce the same partial order over outcomes.

In general the values returned by CPD and KPD can be different. More precisely, the pairs of outcomes for which CPD could give an incorrect contribution to the distance are those that are either incomparable in both CP-nets (in this case CPD could generate an error of $+p$ or $-p$), or that are incomparable in a CP-net and ordered in the other (in this case the CPD error can be +1). To give upper and lower bounds to the error that CPD can make, we study the number of incomparable pairs present in a CP-net. We show that it is polynomial to compute the number of incomparable pairs of outcomes in a separable CP-net (that is, CP-nets with no dependencies among features). Non-separable CP-nets have fewer incomparable pairs of outcomes, since each dependency link eliminates at least one incomparable pair.

Our theoretical bounds are fairly wide. For this reason, we perform an experimental analysis of the relationship between CPD and KTD, which shows that the average error is never more than 10%. We then define a procedure that evaluates the distance between subjective preferences and ethical principles, and makes decisions using the subjective preferences if they are *close enough* to the ethical principles. Otherwise, the procedure moves to less preferred decisions until we find one that is a compromise between the ethical principles and the preferences. We then perform an experimental evaluation showing that the quality of the decision with respect to the subjective preferences does not significantly degrade, i.e., only needs to be moved a short distance in the preference order, when we need compliance with the ethical principles.

Background: CP-nets

CP-nets (Boutilier *et al.* 2004) (for Conditional Preference networks) are a graphical model for compactly representing conditional and qualitative preference relations. They are sets of *ceteris paribus* preference statements (cp-statements). For instance, the cp-statement *"I prefer red wine to white wine if meat is served."* asserts that, given two meals that differ *only* in the kind of wine served *and* both containing meat, the meal with red wine is preferable to the meal with white wine. Formally, a CP-net has a set of features $F = \{x_1, \ldots, x_n\}$ with finite domains $\mathcal{D}(x_1), \ldots, \mathcal{D}(x_n)$. For each feature x_i, we are given a set of *parent* features $Pa(x_i)$ that can affect the preferences over the values of x_i. This defines a *dependency graph* in which each node x_i has $Pa(x_i)$ as its immediate predecessors. An *acyclic* CP-net is one in which the dependency graph is acyclic. Given this structural information, one needs to specify the preference over the values of each variable x for *each complete assignment* on $Pa(x)$. This preference is assumed to take the form of a total or partial order over $\mathcal{D}(x)$. A cp-statement has the general form $x_1 = v_1, \ldots, x_n = v_n : x = a_1 \succ \ldots \succ x = a_m$, where $Pa(x) = \{x_1, \ldots, x_n\}$, $D(x) = \{a_1, \ldots, a_m\}$, and $\succ$ is a total order over such a domain. The set of cp-statements regarding a certain variable X is called the cp-table for X.

Consider a CP-net whose features are A, B, C, and D, with binary domains containing f and $\overline{f}$ if F is the name of the feature, and with the cp-statements as follows: $a \succ \overline{a}$, $b \succ \overline{b}$, $(a \wedge b) : c \succ \overline{c}$, $(\overline{a} \wedge \overline{b}) : c \succ \overline{c}$, $(a \wedge \overline{b}) : \overline{c} \succ c$, $(\overline{a} \wedge b) : \overline{c} \succ c$, $c : d \succ \overline{d}$, $\overline{c} : \overline{d} \succ d$. Here, statement $a \succ \overline{a}$ represents the unconditional preference for $A = a$ over $A = \overline{a}$, while statement $c : d \succ \overline{d}$ states that $D = d$ is preferred to $D = \overline{d}$, given that $C = c$.

A *worsening flip* is a change in the value of a variable to a less preferred value according to the cp-statement for that variable. For example, in the CP-net above, passing from $abcd$ to $ab\overline{c}d$ is a worsening flip since c is better than $\overline{c}$ given a and b. One outcome α is *better* than another outcome β (written $\alpha \succ \beta$) if and only if there is a chain of worsening flips from α to β. This definition induces a preorder over the outcomes, which is a partial order if the CP-net is acyclic.

Finding the optimal outcome of a CP-net is NP-hard (Boutilier *et al.* 2004). However, in acyclic CP-nets, there is only one optimal outcome and this can be found in linear time by sweeping through the CP-net, assigning the most preferred values in the cp-tables. For instance, in the CP-net above, we would choose $A = a$ and $B = b$, then $C = c$, and then $D = d$. In the general case, the optimal outcomes coincide with the solutions of a set of constraints obtained replacing each cp-statement with a constraint (Brafman and Dimopoulos 2004): from the cp-statement $x_1 = v_1, \ldots, x_n = v_n : x = a_1 \succ \ldots \succ x = a_m$ we get the constraint $v_1, \ldots, v_n \Rightarrow a_1$. For example, the following cp-statement (of the example above) $(a \wedge b) : c \succ \overline{c}$ would be replaced by the constraint $(a \wedge b) \Rightarrow c$.

In this paper we want to compare CP-nets while leveraging the compactness of the representation. To do this, we consider profile (P, O), where *P* is a collection of n CP-nets (whose graph is a directed acyclic graph (DAG)) over m common variables with binary domains and *O* is a total order over these variables. We require that the profile is O-legal (Lang and Xia 2009), which means that in each CP-net, each variable is independent to all the others following in the ordering O. Given a variable X_i the function $flw(X_i)$ returns the number of variables following X_i in O.

Since every acyclic CP-net is satisfiable (Boutilier *et al.* 2004), we compute a distance among two CP-nets by comparing a linearization of the partial orders induced by the two CP-nets. In this paper, we consider the linearization generated using the algorithm described in the proof of Theorem 1 of (Boutilier *et al.* 2004) and reproduced below as Algorithm 1. This algorithm works as follows: Given an acyclic CP-net A over n variables and a ordering O to which the A is O-legal, we know there is at least one variable with no parents. If more than one variable has no parents, then we choose the one that comes first in the provided ordering O; let X be such a variable. Let $x_1 \succ x_2$ be the ordering over $Dom(X)$ dictated by the cp-table of X. For each $x_i \in Dom(X)$, construct a CP-net, N_i, with the $n - 1$ variables $V - X$ by removing X from the initial CP-net, and for each variable Y that is a child of X, revising its CPT by restricting each row to $X = x_i$. We can construct a preference ordering $\succ_i$ for each of the reduced CP-nets N_i. For each N_i recursively identify the variable X_i with no parents and construct a CP-net for each value in $Dom(X_i)$ following the same algorithm until a CP-net have variables. We can now construct a preference ordering for the original network A by ranking every outcome with $X = x_i$ as preferred to any outcome with $X = x_j$ if $x_i \succ x_j$ in CPT(X). This linearization, which we denote with $LexO(A)$, assures that ordered pairs in the induced partial order are ordered the same in the linearization and that incomparable pairs are linearized using the cp-tables.

In Algorithm 1, $CPT_{A,o}(v)$ returns the ordered values of variable v in CP-net A, given a partial assignment o to a subset of variables. This linearization, which we denote with $LexO(A, O)$, where A is a CP-net and O an O-legal order over the features of A, enforces that ordered pairs in the induced partial order are ordered the same in the linearization and that incomparable pairs are linearized using the cp-tables.

Algorithm 1 Linearization of a Partial Order induced by a CP-net A

```
function LexO($A, O, Lin = [], o = None$)   ▷ Where $A$ is a CP-net, $O$ is the O-legal order on $A$, $Lin$ is the (initially empty) linearization computed by the function, and $o$ is an outcome (initially none).
    if $O = Null$ then
        $Lin.append(o)$
        return $Lin$
    end if
    $v = pop(O)$
    for $value \in CPT_{A,o}(v)$ do
        $temp = o + value$
        $Lin = LexO(A, O, Lin, temp)$
    end for
    return $Lin$
end function
```

A CP-net Distance Function

In what follows we will assume that all CP-nets are acyclic and in minimal (non-degenerate) form, i.e., all arcs in the dependency graph have a real dependency expressed in the cp-statements, see the extended discussion in (Allen *et al.* 2017; 2016). The following definition is an extension of the (Kendall 1938) τ (KT) distance with a penalty parameter p defined for partial rankings by (Fagin *et al.* 2006).

Definition 1. *Given two CP-nets A and B inducing partial orders P and Q over the same set of outcomes U:*

$$KTD(A, B) = KT(P, Q) = \sum_{\forall i,j \in U, i \neq j} K^p_{i,j}(P, Q) \quad (1)$$

where i and j are two outcomes with $i \neq j$, we have:

1. *$K^p_{i,j}(P, Q) = 0$ if i, j are ordered in the same way or they are incomparable in both P and Q;*
2. *$K^p_{i,j}(P, Q) = 1$ if i, j are ordered inversely in P and Q;*
3. *$K^p_{i,j}(P, Q) = p$, $0.5 \leq p < 1$ if i, j are ordered in P (resp. Q) and incomparable in Q (resp. P).*

In the previous definition we choose $p \geq 0.5$ to make $KTD(A, B)$ a distance function, indeed if $p < 0.5$ the distance does not satisfy the triangle inequality. We also exclude $p = 1$ so that there is a penalty for two outcomes being considered incomparable in one and ordered in another CP-net. This allows us, assuming O-legality, to define for each CP-net a unique most distant CP-net.

Proposition 1. *Given two acyclic CP-nets A and B that are not O-legal, deciding if $KTD(A, B) = 0$ cannot be computed in polynomial time unless $P = NP$.*

The NP-complete problem of checking for equivalence for two arbitrary CP-nets (Santhanam *et al.* 2013), i.e., deciding if two CP-nets induce the same ordering, can be reduced to the problem of checking if their KTD distance is 0. That is, if we had a polynomial time algorithm for deciding if $KTD(A, B) = 0$ then we could decide the equivalence problem for acyclic CP-nets. We know from (Boutilier *et* *al.* 2004) that dominance testing for max-δ-connected CP-nets, that is CP-nets where the maximum number of paths between two variables is polynomially bounded in the size of the CP-net is NP-complete. We know that O-legal, acyclic CP-nets are a class of max-δ-connected CP-nets because the O-legality constraint means that there are only a maximum of $n-2$ paths between two nodes. However, this does not necessarily mean that the equivalence question is automatically hard. As we will see, our lower bound can actually be used to check equivalence for acyclic, O-legal CP-nets.

Since the question of dominance is closely related to that of distance, the complexity of computing KTD for O-legal CP-nets remains an important open question that we conjecture to be intractable. Due to this likely intractability we will define another distance for CP-nets which can be computed efficiently directly from the CP-nets without having to explicitly compute the induced partial orders. This new distance is defined as the Kendal Tau distance of the two $LexO$ linearizations of the partial orders.

Definition 2. *Given two O-legal CP-nets A and B, with m features, we define:*

$$CPD(A,B) = KT(LexO(A), LexO(B)) \qquad (2)$$

We show that CPD is a distance over O-legal CP-nets.

Theorem 1. *Function CPD(A,B) satisfies the following properties:*

1. $CPD(A,B) \geq 0$*;*
2. $CPD(A,B) = CPD(B,A)$*;*
3. $CPD(A,B) \leq CPD(A,C) + CPD(C,B)$*.*
4. $CPD(A,B) = 0$ *if and only if* $A = B$*;*

Proof. Properties 1-3 are directly derived from the fact that KTD is a distance function over total orders. Let us now focus on property 4. In our context, $A = B$ if and only if they induce the same partial order. It is, thus, obvious that if $A = B$ then $CPD(A,B) = 0$ since $LexO(A) = LexO(B)$. Let us now assume that $A \neq B$. Thus A and B induce different partial orders. In principle, what could happen is that one partial order is a subset of the other. In such a case they would have the same $LexO$ linearizations and it would be the case that $CPD(A,B) = 0$, despite them being different. We need to show that this cannot be the case if A and B are O-legal. Let us first assume that A and B have the same dependency graph but that they differ in at least one ordering in one CP-table. It is easy to see that in such a case there is at least one pair of outcomes that are ordered in the opposite way in the two induced partial orders. Assume that A and B have a different dependency graph. Due to O-legality it must be that there is a least an edge which is present, say, in A and missing B. In this case by adding a non-redundant dependency we are reversing the order of at least two outcomes. □

We will now show how $CPD(A,B)$ can be directly computed from CP-nets A and B, without having to compute the linearizations. The computation comprises of two steps. The first step, which we call, normalization, modifies A and B so that each feature will have the same set of parents in both CP-nets. This means that each feature will have in both normalized CP-nets a CP-table with exactly the same number of rows corresponding each to the same assignment to its parents. The second step, broadly speaking, computes the contribution to the distance of each difference in the CP-table entries. We describe each step in turn.

Step 1: Normalization. Consider two CP-nets, A and B over m variables $V = \{X_1, \dots, X_m\}$ each with binary domains. We assume the two CP-nets are O-legal with respect to a total order $O = X_1 < X_2 < \cdots < X_{m-1} < X_m$. We note that O-legality implies that the X_i can only depend on a subset of $\{X_1, \dots, X_{i-1}\}$

Each variable X_i has a set of parents $Pa_A(X_i)$ (resp. $Pa_B(X_i)$) in A (resp. in B), and is annotated with a conditional preference table in each CP-net, denoted $CPT_A(X_i)$ and $CPT_B(X_i)$.

We note that, in general we will have that $Pa_A(X_i) \neq Pa_B(X_i)$. However, it is easy to extend the two CP-nets so that in both X_i will have the same set of parents $Pa_A(X_i) \cup Pa_B(X_i)$. This is done by adding redundant information to the CP-tables, which does not alter the induced ordering.

For example, let us consider $CPT_A(X_i)$, then we will add $2^{Pa_A(X_i) \cup Pa_B(X_i)} - 2^{Pa_A(X_i)}$ copies of each original row to $CPT_A(X_i)$, that is, one for each assignment to the variables on which X_i depends in B but not in A. After this process is applied to all the features in both CP-nets, each feature will have the same parents in both CP-nets and its CP-tables will have the same number of rows in both CP-nets. We denote with A' and B' the resulting CP-nets.

We note that normalization can be seen as the reverse process of CP-net reduction (Apt *et al.* 2008) which eliminates redundant dependencies in a CP-net.

Step 2: Distance Calculation Given two normalized CP-nets A and B, let $diff(A,B)$ represent the set of CP-table entries of B which are different in A and let $var(i) = j$ if CP-table entry i refers to variable X_j. Moreover, let $m = |V|$ and $flw(X)$ denote the number of features following X in order O. Let us define the two following quantities:

$$nSwap(A,B) = \sum_{j \in diff(A,B)} 2^{flw(var(j)) + (m-1) - |Pa_B(var(j))|} \qquad (3)$$

which counts the number of inversions that are caused by each different table entry and sums them up.

Theorem 2. *Given two normalized CP-nets A and B, we have:*

$$CPD(A,B) = nSwap(A,B) \qquad (4)$$

We provide an example of how a difference in a CP-table entry affects the $LexO$ linearization.

Example 0.1. *Consider a CP-net with three binary features, A, B, and C, with domains containing f and $\overline{f}$ if F is the name of the feature, and with the cp-statements as follows: $a \succ \overline{a}$, $b \succ \overline{b}$, $c \succ \overline{c}$. A linearization of the partial* *order induced by this CP-net can be obtained by imposing an order over the variables, say Let variable ordering $O = A \succ B \succ C$. The $LexO(A)$ is as follows:*

$$\overbrace{\overbrace{abc \succ ab\overline{c}}^{B1Zone} \succ \overbrace{a\overline{b}c \succ a\overline{b}\overline{c}}^{B2Zone}}^{A1Zone} \succ \overbrace{\overbrace{\overline{a}bc \succ \overline{a}b\overline{c}}^{B3zone} \succ \overbrace{\overline{a}\overline{b}c \succ \overline{a}\overline{b}\overline{c}}^{B4zone}}^{A2zone}$$

Now, consider changing only the cp-statement regarding A to $\overline{a} \succ a$. Then, the linearization of this new CP-net can be obtained by the previous one by swapping the first outcome in the $A1zone$ with the first outcome in the $A2zone$, the second outcome in the $A1zone$ with the second outcome in the $A2zone$ and so on. Moreover, the number of swaps is directly dependent on the number of variables that come after A in the total order.

From Theorem 2 we can see that $0 \leq CPD(A, B) \leq 2^{m-1}(2^m - 1)$, where m is the number of features. In particular:

- $CPD(A, B) = 0$ when the two CP-nets have the same dependency graph and cp-tables and so they are representing the same preferences;
- $CPD(A, B) = 2^{m-1}(2^m - 1)$ when the two CP-nets have the same dependency graph but cp-tables with reversed entries, so they are representing preferences that are opposite to each other.

Notice that variables with different cp-statements in the representation give more value to the distance if they come first in the total order: the value decreases as the position in the total order increases. For instance it is easy to prove that if the cp-statement of the first variable in the total order differs, than $CPD \geq 2^{m-2}(2^m - 1)$.

Supporting Ethical Decisions

Ethical principles are modelled via a CP-net, say S, and an individual models her preferences via another CP-net, say B. We assume that these two CP-nets have the same features.

Of course this is a restriction and in general we think the features of these two CP-nets can overlap but not necessarily be the same. We are studying what happens when the two sets of features do not coincide. But for the purpose of this paper we will assume they do coincide.

Given the ethical principles and the individual's preferences, we need to guide the individual in making decisions that are not too unethical. To do this, we propose to proceed as follows:

1. We set two distance thresholds: one between CP-nets (ranging between 0 and 1), and another one between decisions (ranging between 1 and n).
2. We check if the two CP-nets A and B are less distant than t_1. In this step, we use CPD to compute the distance.
3. If so, the individual is allowed to choose the top outcome of his preference CP-net.
4. If not, then the individual needs to move down its preference ordering to less preferred decisions, until he finds one that is closer than t_2 to the optimal ethical decision. This is a compromise decision between what the preferences say and what the ethical principles recommend.

Empirical Analysis

We divide the empirical evaluation in two parts. Firstly, we evaluate the performances of the CPD distance by checking running time and deviation from the exact KTD distance. The first part of the experiments shows that in terms of computation time and error rate, our approximation performs extremely well. The second part of our experiments focuses on the ethical perspective. We show how the distance can be used in an ethical scenario to evaluate how much an individual decision maker deviates from an adopted ethical principle modeled as a CP-net.

Ethical Scenario

Given an ethical principle and the preference of an individual, both encoded as CP-nets, we want to understand if following the preferences will lead to an ethical action. Since in this scenario individuals want to act ethically, firstly the individual determines whether she can use her most preferred choice by checking if her CP-net is "sufficiently close" to the ethical CP-net. If these two CP-nets are farther apart than some threshold t_1, then we proceed down the preference ordering till we find a decision that is sufficiently close to the optimal ethical decision, according to another threshold t_2.

We represent the ethical principles with a CP-net A and the individual's preferences with a CP-net B, and we assume that these two CP-nets have the same features. We judge that the individual is acting ethically if $CPD(A, B) \leq t_1$. If yes, the individual knows that her preferences are pretty ethical and she can choose the best outcome induced by her CP-net.

If instead $CPD(A, B) > t_1$, we compute how many worsening flips we need to apply to her best decision (according to her preferences) to get to a decision that is closer than t_2 flips from the optimal ethical decision.

This empirical analysis is run varying n, t_1 and t_2, where n is the number of features, and t_1 and t_2 are the tolerances. We run experiments varying the number of features $2 \leq n \leq 8$. For each value of n we vary $t_1 \in \{0, 0.1, 0.2, 0.4, 0.8\}$. Low values of t_1 represents scenarios where the tolerance is absent or low. This means that, in order for a decision maker to take their first choice, they should have preferences very close to the ethical principle. Larger values of t_1 model less strict ethics, where people have more freedom of choice. For each value of n and t_1, we vary the value of t_2 ($2 \leq t_2 \leq (n + 2)/2$). This again represents scenarios where the freedom of individuals vary.

Given the values of n, t_1,, and t_2 we generate 1000 pairs of CP-nets (A, B) from a uniform distribution using the software described by (Allen *et al.* 2017; 2016). We compared values of the approximate CPD distance with the real KTD distance. This shows us how many times CPD is wrong and how much individuals need to sacrifice of their preferences in order to be ethical. We consider and report the following cases which represent the *confusion matrix* of our experiment:

1. True Positive (TP): $CPD(A,B) \leq t_1$ and $KTD(A,B) \leq t_1$. In this case, individual preferences are close to the ethical principles and decision makers choose their best alternative;
2. True Negative (TN): $CPD(A,B) > t_1$ and $KTD(A,B) > t_1$. In this case, individual preferences are not close to the ethical principles and the decision makers must find a compromise;
3. False Positive (FP): $CPD(A,B) \leq t_1$ and $KTD(A,B) > t_1$. In this case, erroneously, individuals think they are acting ethically and consequently choose their best alternative even though it is not ethical;
4. False Negative (FN): $CPD(A,B) > t_1$ and $KTD(A,B) \leq t_1$. In this case, erroneously individuals think they are not acting ethically and they select a compromise decision even though they could select their top preferred decision.

The number of $TP + TN$ gives an idea of the accuracy of the distance; the higher this value, the higher confidence individuals can have in using the approximation of the distance to understand whether they are ethical or not.

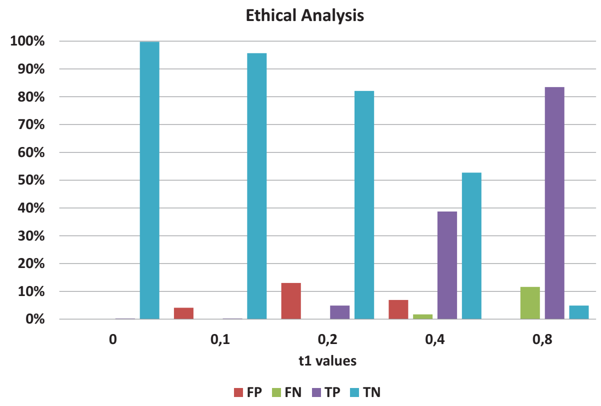


Figure 1: Percentage of TP, TN, FP, FN: the chart reports the number of cases for which CPD and KTD agree, or not, on the comparison based on the tolerance t_1. This gives an idea of the accuracy of the approximated distance.

Figure 1 shows the confusion matrix for $n = 7$ and $t_2 = 4$ while varying t_1. Notice that, as expected, when the tolerance t_1 is null or low, e.g., $t_1 = 0$ or $t_1 = 0.2$, individuals can almost never select their their best choice. Indeed, for $t_1 = 0$ the percentage of True Positives (purple bar) is close to 0% while for $t_2 = 0.2$ the percentage of True Positive is around 5%. This means that the decision makers preferences must be close to the ethical principle in order to have the freedom to choosing their best choice. Instead, when the tolerance is higher, they have more freedom to choose what they like. For example, with $t_1 = 0.4$, the percentage of True Positives (purple bar) is close to 40% while for $t_1 = 0.8$ it is more than 80%.

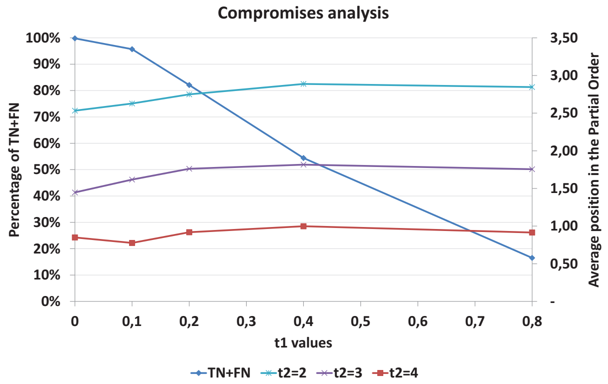


Figure 2: Compromises analysis: the charts reports a comparison between the number of times that individuals have preferences which are not close to the ethics and for which they have to look for a compromise and the quality of the compromise in terms of distance from their best choice.

The next important question is: What happens when individuals cannot choose their first choice and have to look for another one which is closer to the ethical principles? Figure 2 reports the percentage of cases in which individuals have to find a compromise because their preferences are not close to the ethical principles, according to t_1,. For these cases we quantify the amount of compromise in terms of positions in the induced partial order. As before, when the tolerance is strict, an individual has to look for a compromise nearly every time. It is interesting to notice that the amount of compromise varies based on the value of t_2 and seems to be not influenced by t_1. This is quite natural, when $t_2 = 4$ it means that the individual has to find a choice that is in the top five positions of the ethical ordering in order to reach a compromise. This means that such a choice, on average, is in the first two positions of the individual's preference (red line in figure). The lower the value of t_2, the harder it becomes for the individual to find an ethical decision, and she has to descend down her preference order, on average, up to the fourth position to find an acceptable alternative.

Conclusions

In order to model and reason with both preferences and ethical principles in a decision making scenario, we have proposed a notion of distance between CP-nets, providing both a theoretical study and an experimental evaluation of its properties. We show that our approximation is both accurate in practice and efficient to compute.

Several extensions to our setting can be considered for the future. Indeed, we have made some assumptions on the two CP-nets for which we can compute the distance, that would be useful to relax. First, the two CP-nets over which we define the CPD distance have the same features, and with the same domains, but can differ in their dependency structure and CP-tables. It is important to also cover the case of CP-nets that may have different features and domains. Moreover, we have also assumed the two CP-nets are O-legal, that is, there is a total order of the CP-nets features that is compatible with the dependency links of both CP-nets. Intuitively, this means that the preferences are the ethical principles are not indicating completely opposite priorities. However, there could be situations where this is actually the case, and it is important to know how to combine preferences and ethical principles also in this case.

References

T.E. Allen, M. Chen, J. Goldsmith, N. Mattei, A. Popova, M. Regenwetter, F. Rossi, and C. Zwilling. Beyond theory and data in preference modeling: Bringing humans into the loop. In *Proceedings of the 4th International Conference on Algorithmic Decision Theory (ADT)*, 2015.

T.E. Allen, J. Goldsmith, H.E. Justice, N. Mattei, and K. Raines. Generating CP-nets uniformly at random. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

T. E. Allen, J. Goldsmith, H. E. Justice, N. Mattei, and K. Raines. Uniform random generation and dominance testing for cp-nets. *Journal of Artificial Intelligence Research*, 59:771–813, 2017.

Krzysztof R. Apt, Francesca Rossi, and Kristen Brent Venable. Comparing the notions of optimality in cp-nets, strategic games and soft constraints. *Ann. Math. Artif. Intell.*, 52(1):25–54, 2008.

Jean-François Bonnefon, Azim Shariff, and Iyad Rahwan. The social dilemma of autonomous vehicles. *Science*, 352(6293):1573–1576, 2016.

Craig Boutilier, Ronen Brafman, Carmel Domshlak, Holger Hoos, and David Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

Ronen I. Brafman and Yannis Dimopoulos. Extended semantics and optimization algorithms for CP-networks. *Computational Intelligence*, 20(2):218–245, 2004.

Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. Preference handling in combinatorial domains: From AI to social choice. *AI Magazine*, 29(4):37–46, 2008.

Yann Chevaleyre, Frédéric Koriche, Jérôme Lang, Jérôme Mengin, and Bruno Zanuttini. Learning ordinal preferences on multiattribute domains: The case of CP-nets. In *Preference Learning*, pages 273–296. Springer, 2011.

Vincent Conitzer, Jérôme Lang, and Lirong Xia. Hypercubewise preference aggregation in multi-issue domains. In *22nd*, pages 158–163, 2011.

David Copp. *The Oxford Handbook of Ethical Theory*. Oxford University Press, 2005.

C. Cornelio, J. Goldsmith, N. Mattei, F. Rossi, and K.B. Venable. Updates and uncertainty in CP-nets. In *Proceedings of the 26th Australasian Joint Conference on Artificial Intelligence (AUSAI)*, 2013.

C. Cornelio, U. Grandi, J. Goldsmith, N. Mattei, F. Rossi, and K.B. Venable. Reasoning with PCP-nets in a multi-agent context. In *Proceedings of the 14th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2015.

C. Domshlak, E. Hüllermeier, S. Kaci, and H. Prade. Preferences in AI: An overview. 175(7):1037–1052, 2011.

Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D. Sivakumar, and Erik Vee. Comparing partial rankings. *SIAM J. Discret. Math.*, 20(3):628–648, March 2006.

J. Goldsmith, J. Lang, M. Truszczyński, and N. Wilson. The computational complexity of dominance and consistency in CP-nets. *Journal of Artificial Intelligence Research*, 33(1):403–432, 2008.

M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

Jerome Lang and Lirong Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical Social Sciences*, 57(3):304–324, 2009.

A. Loreggia, N. Mattei, F. Rossi, and K.B. Venable. On the distance between CP-nets. In *Proceedings of the 17th International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2018.

N. Mattei and T. Walsh. PrefLib: A library for preferences, HTTP://WWW.PREFLIB.ORG. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT)*, 2013.

N. Mattei, M. S. Pini, F. Rossi, and K. B. Venable. Bribery in voting with CP-nets. *Annals of Mathematics and Artificial Intelligence*, 68(1–3):135–160, 2013.

G. Pigozzi, A. Tsoukiàs, and P. Viappiani. Preferences in artificial intelligence. 77:361–401, 2015.

F. Rossi, K.B. Venable, and T. Walsh. *A Short Introduction to Preferences: Between Artificial Intelligence and Social Choice*. Morgan and Claypool, 2011.

Ganesh Ram Santhanam, Samik Basu, and Vasant Honavar. Verifying preferential equivalence and subsumption via model checking. In Patrice Perny, Marc Pirlot, and Alexis Tsoukiàs, editors, *Algorithmic Decision Theory - Third International Conference, ADT 2013, Bruxelles, Belgium, November 12-14, 2013, Proceedings*, volume 8176 of *Lecture Notes in Computer Science*, pages 324–335. Springer, 2013.

Amartya Sen. *Choice, Ordering and Morality*. Blackwell, Oxford, 1974.

Judith Jarvis Thomson. The trolley problem. *The Yale Law Journal*, 94(6):1395–1415, 1985.

Importance of Contextual Knowledge in Artificial Moral Agents Development

Rafal Rzepka, Kenji Araki
Graduate School of Information Science and Technology
Hokkaido University, Kita-ku, Kita 14, Nishi 8
060-0814 Sapporo, Japan

Abstract

In this paper we underline the importance of knowledge in artificial moral agents and describe our experience-focused approach which could help existing algorithms go beyond proofs of concept level and be tested for generality and real-world usability. We point out the difficulties with implementation of current methods and their lack of contextual knowledge hindering simulations in more realistic, every-day life situations. The idea is to prioritize resources for predictions and the process of automatic knowledge acquisition for an oracle to be used by moral agents, both human and artificial.

Introduction

Value alignment problem has recently gained the attention among artificial intelligence researchers, philosophers and non-specialists. Nick Bostrom's book "Superintelligence" (Bostrom 2014) has popularized the topic and the potential dangers of high-level autonomy machines became widely discussed, also by influential figures as Stephen Hawking, Bill Gates or Elon Musk. Although the discussion has lasted for years and many possible solutions have been proposed, a universally moral machine is still far from reality. One of the major problems is the fact that universal code of ethics is hard (or impossible) to establish due to the cultural differences and the influences of other bigger and smaller contextual variations. For that reason the Artificial Intelligence researchers are in a difficult position when they try to create an ethical decision making system which could help alleviating worries about the future of AI.

Hypothesis

Our hypothesis is that knowledge acquisition field should be as important for building unprejudiced systems as the very process of creating them. Top-down approaches are difficult to implement (what exactly does one mean by "do not harm"?) and in bottom-down approaches the decision process generates output which origin is often difficult to explain. However, both strategies could benefit from gathered and processed[1] real world descriptions and become more easily testable and expandable. Because we work, among others, on dialog systems, basically without any input restrictions, workable but sufficiently general methods for moral evaluation are necessary at the very moment. For this reason we aim at solutions not for hypothetical super-intelligent agents dealing with famous moral dilemmas but for existing systems which must avoid learning failures like Microsoft's Tay bot that was tricked to praise Hitler. On the other hand, we also are aware about pitfalls of gathering knowledge without quality considerations. In this paper we would like to emphasize the importance of rich examples or real-world situations. To the authors' best knowledge all existing implementations of artificial moral agents (AMAs) deal only with toy applications (prototypes) and very specific / limited tasks. We think the shortage of contextual data is one of the main reasons restraining AI systems from becoming more general, expandable and testable. Knowledge in these systems is manually crafted making them less realistic and difficult to be implemented in real-world, everyday applications.

Existing Approaches to Machine Ethics

Variety of possible solutions for achieving ethical machines have been proposed and one of the latest survey[2] of existing methods is given in (Pereira and Saptawijaya 2016). Authors of this book divide the approaches into two realms – one dealing with individual ethical instances and second describing collective morality, which combines game theory (Conitzer et al. 2017) and findings of the evolutionary psychology. They introduce their approach using logic programing for individual moral agents and propose methods for bridging both realms. Logic-based methods, for example by formalizing ethical codes with deontic logic of (Bringsjord, Arkoudas, and Bello 2006) are probably the most popular and machine learning (Anderson, Anderson, and Armen 2006) is not used widely as it is often believed that machine ethics cannot be based on predicting how to do the right thing. We partially disagree with (Pereira and Saptawijaya 2016) claiming that *the community should be well aware that such present day learning is inadequate for*

[1] For instance automatically annotated with probability, polarity or utility estimations.

[2] Due to the limited space we only mention main types of approaches; our proposal refers to probably all AMAs but is different in focusing on knowledge rather than algorithms.

general machine morality. Only small, circumscribed, well-defined domains have been susceptible to rule generation through machine learning. Rules are all important for moral explanation, justification and argumentation. We think rules can be extrapolated from fuzzy observations and we believe the observations helped humans greatly in creating ethics (as well as language, mathematics or logics). We return to this theme several times in this paper as the knowledge (result of observations) is one of the two cores of our approach.

On the other hand, we are also aware that learning from Big Data in the spirit of purely statistical and probabilistic calculations is also flawed, risky and the agent's reasoning often cannot be explained. However, we think that methods like deep learning can enrich the textual data which lacks tacit knowledge. We also see a problem common to probably all approaches, including cognitive architectures (Bretz and Sun 2017) – the need of creating correct data sets and (preferably contextual) knowledge bases. Their manual creation / annotation is costly and impracticable when all even only the most probable situations that an agent may face are needed to be considered[3]. To address this problem, methods for automatic acquisition of moral rules e.g. by human-machine cooperation with Inverse Reinforce Learning (Ng, Russell, and others 2000; Hadfield-Menell et al. 2016) were suggested. However, they, silimarly to the "seed AI"-like approaches, also seem unrealistic because teaching (supervising) an agent to deal with complex cases in changing environments could take very long time and the AMA would be influenced by one supervisor's experiences and his or her preferences. Thorough the scrutiny of formal methods and shallow but wide stochastic approaches can help each other or even be integrated into more holistic systems for example using probabilistic methods like Bayesian interference (Tenenbaum et al. 2011). But before that, at least in our opinion, it seems necessary to provide more structured crowd-based contextual data which could allow:

- discovering causes and effects
- calculating probabilities
- forming and dissolving abstract knowledge
- simulating real world situations
- testing existing and new moral agents

In the next section we describe our approach which discovers causes and effects for the moral judgement task. After that we present our idea of expanding the existing ontologies to deal with concepts as stories, the need of controlling data credibility and the importance of language itself. In the last part of this paper we answer several questions that often appear when discussing our approach with other researchers.

Knowledge-First Approach

Our proposal is to shortly go back to the point in our evolution when no theories of ethics were yet formulated. We assume that empathic circuitry in our brains, together with the capabilities to observe the world and to communicate with peers ignited codification of our sense of justice which keeps changing throughout the ages. The idea is to simulate this process (and test our hypothesis) by first creating conditions for discovering contextual dependencies that influence moral load of given states and acts. These conditions are currently reduced to a) unstructured knowledge in natural language b) agent's capability to guess a polarity (positive or negative) of concepts (acts or states).

Source Knowledge

As mentioned before, although the broad world knowledge seems to be an obvious ingredient of moral reasoning, it is widely ignored by the creators of Artificial Moral Agents. To show that it is not only useful but crucial in machine ethics we utilize various text resources like blog corpus, Twitter corpus, Aozora book repository (we mostly work with Japanese language), chat logs, etc. which contain billions of words. Basically matching concepts and the natural language processing is performed of a limited context of on, two or three sentences (sentence with a concept being analyzed, previous sentence containing possible reasons and following sentence with possible consequences).

Polarity Calculation

For time being we utilize sentiment analysis methods to help our systems asses consequences. The initial idea is presented in (Rzepka and Araki 2005) and more technical details are given in (Rzepka and Araki 2012) and (Rzepka and Araki 2015). The simplest method for this task utilizes lexicons of negative and positive words. For example, if most of human experiences with "stealing a car" described in text resources cooccurred with negative lexicon words, the polarity of the concept becomes morally negative. Except emotion-related phrases we also created a lexicon based on Kohlberg's stages of moral development (praising / reprimanding, awarding / punishing, etc.) to extend recognition to legal consequences (if an act ended in doer's arrest it is more likely that the act was not moral).

Precision of Moral Estimation

The latest experiments (Rzepka and Araki 2017) show that our simplistic approach is able to achieve almost 85.7% agreement with human subjects. However, the results showed that mere size of knowledge base does not equate to better ethical judgement. Not only different automatic polarity estimation methods must be tested, also the credibility of the sources require investigation. We elaborate on this problem and propose solutions in later sections. It also must be noted that the experiments were performed with concepts and many of them strongly depend on wider context. The input is basically unrestricted when it comes to the topic but longer concepts decrease the chance of finding sufficient number of examples. For instance *driving* should be recognized as neutral, *driving after drinking* as negative, but *driving with a baby unbuckled after home party at friends house on the hill* cannot be found in the given input form, therefore recognizing, abstracting and weighting concepts within the

[3] In logic-based approaches knowledge is limited to a given task, usually a single dilemma in very restricted environment.

input becomes necessary. Unfortunately, automizing these tasks is rather difficult without sufficient set of reliable examples from which e.g. a concept's importance can be calculated. This is one of the reasons we are currently preparing the ontology of concepts discussed later in this paper.

Tests with Embodied System

Many researchers draw attention to the importance of embodiment in moral behavior (Trappl 2015) and need concrete testing decision-making algorithm in action (Arnold and Scheutz 2016). To see how our text knowledge-based approach works in the real world, we implemented our method on a Roomba robotic vacuum cleaner (Takagi, Rzepka, and Araki 2011). Users were allowed to communicate freely with the device through Twitter. The robot had its name ("Roomba") and function (variants of the verb "to clean") hardcoded, and its mission was to make a user happy without violating common sense, which is the motto of our approach. The system, with knowledge base limited only to Twitter corpus worked surprisingly well and the robot was able to propose its help even if no straightforward command was given. For instance, "this room is a mess" has triggered negative reactions and Twimba (the name of our system) found by simple search that people deal with this problem by cleaning which was its capability. On the other hand, when one talks about a "dirty look", the robot does not react, because it deals with concepts, not single words. Not caring about even very small contexts, although common in various machine learning methods, showed us clearly that deeper and more careful approach is necessary. Naturally it was hard for a vacuum cleaner to violate common sense, but "knowing" its name and its only function helped it to refuse cleaning a bathtub, only because no examples of Roombas cleaning bathtubs were found in the knowledge base.

Other Characteristics and Possibilities

As showed above, the sophistication of moral behavior may increase with machine capabilities but does not seem to be limited to embodied agents. Obviously the more actions a machine can perform, the more dangerous it can become, but e.g. chat systems with purpose like the second language acquisition tutor (Nakamura et al. 2017) have to deal with abstract concepts and utilize their "talking" capability that conveys meaning which can be directly and indirectly harmful to the user. For example, an artificial tutor reacting positively to a bullying statement is not only unnatural but also may negatively influence adolescent users.

When implemented in a dialog system, our method needs to support explaining its judgements which is an important functionality for an AI system (Core et al. 2006). Explainable AI needs linguistic skills and the reasoning should be clear to any user. Simplicity of the current algorithm and dealing only with natural language makes it relatively easy to generate explanations how a given judgement was performed. In case of our majority voting strategy, it is currently enough to use only four output templates: a) "It's moral because majority (X%) of cases had positive consequences", b) "It's immoral because majority (Y%) of cases had negative consequences", c) "It's problematic" and d) "Not enough data". Examples of observations can be also easily added. We have tested different majority thresholds and 60-70% level seems to be most effective (Rzepka and Araki 2017). The non-decisive middle area when roughly half consequences were recognized as good and half as bad ("problematic" output) constitute a safety valve (Rzepka and Araki 2005). It contains concepts like abortion or euthanasia and it advised to program a system with our algorithm to avoid actions and strong statements when even people are not sure about the outcomes. To allow our method to handle such cases and be able to perform ethical judgement and decrease "Not enough data" outputs, again more contextual knowledge is needed.

Toward the Ontology of Contexts

Current Knowledge Bases

Current knowledge bases are stored in various formats but usually they can be represented in a flat and solid, cross-linked structure like hypertext (Wikipedia, DBpedia, BabelNet, etc.) which links terms with other terms, categories or definitions. Ontologies (semantic nets) like CyC (Lenat and Guha 1989) or ConceptNet (Speer and Havasi 2012) try to connect more abstract, commonsensical concepts, but they do not contain longer chains of consecutive concepts which could form, for instance, a Schankian script (Schank and Abelson 1977). Therefore there is a gap between such knowledge bases that cover small chunks of knowledge and just raw text which very often describe much bigger contexts but are incomplete and/or noisy. We treat moral decision making as a subtask of the common sense processing. It requires processing extendable / shrinkable data chunks that constantly change their size and density depending on the stream of information (linguistic in our case). This information always changes as time moves forward and elements of environment alter, but if an apple changes color to brown, it does not mean a concept of apple like *HasProperty* changes from "sweet" to "rotten".

Expanding Number of Concepts and Their Relations

We are currently experimenting with combining existing concepts (from ConceptNet) into longer chunks of possible chains by confronting them with the blog corpora. Several techniques are required for cleaning up the text, recognizing semantic roles, tackling with anaphora resolution, double negations and other NLP-related tasks. Because the Japanese ConceptNet is not big enough, we currently work on expanding it (Krawczyk, Rzepka, and Araki 2016) and checking its quality (Shudo, Rzepka, and Araki 2016). The idea of data being used to supervising other data is not new, it is called distant supervision (Mintz et al. 2009) where the data replaces human in learning or other tasks usually requiring human's assistance. In Figure 3, we show how the text knowledge itself can be useful in both expanding the knowledge base and supervising any machine learning algorithm giving positive and negative feedback from polarity calculation module. Simultaneously we are trying to acquire new

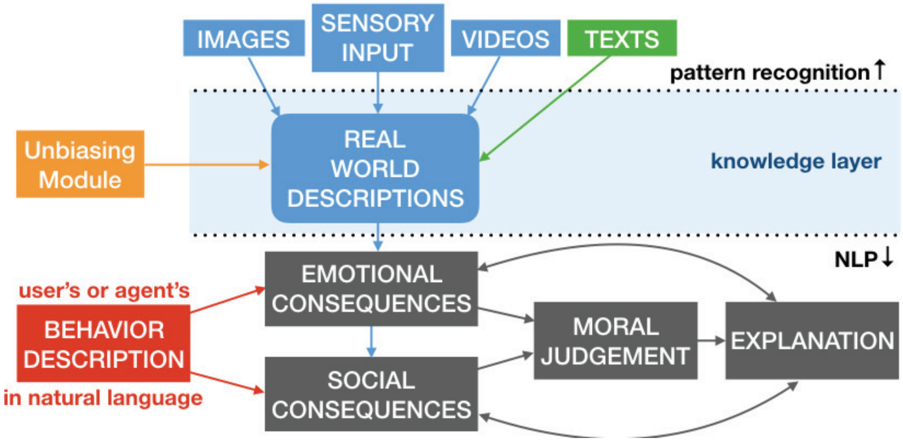


Figure 1: Three layers of language-based moral judgement allowing understandable explanation of ethical choices calculated from polarity of possible consequences).

concepts and their relations with grammar rules and linguistic information like part of speech (see Figure 2). But naturally the biggest difficulties with natural language lay not on the lexical layer, but as we show below, on the semantic level.

Enriching Context with Automatic Descriptions

Another important and unanswered question in common sense knowledge acquisition is how to provide machines with tacit knowledge which is obvious for us thanks to our sensory input and is rarely expressed in language. As we showed in the Figure 1, we believe that advances in pattern recognition will be able to at least partially tackle this problem with methods like deep learning which has already had some successes in automatically describing images in natural language (Vinyals et al. 2015). Currently[4] we simulate sensory input with text-mining techniques (Rzepka, Mitsuhashi, and Araki 2016), but let us assume the progress in pattern recognition (machine learning on constantly growing data) has reached the human level without the massive and costly annotated data. Every image or video available can be described in a natural language in detail and every sentence in written text can be flawlessly parsed. The speed of access and analysis naturally surpasses human capabilities. Our hypothesis is that just because a machine can refer to more experiences (cases, contexts, regulations, etc.) than we can, it is theoretically possible for the machine to generate more fair judgement even than ethicists or judges. Moreover, if programmers ensure that the moral judgement algorithm is not prone to biases (or at least is less biased than most of us), an agent could become an important advisor for human or robotic users. We discuss such a possibility of ideal advising oracle in the last part of this paper.

Credibility Problem

Internet is a source of countless examples of knowledge which is simply wrong. Darker side of human nature reveals itself with spams, scams, flame wars, trolling, conspiracy theories, fake news and so on. Our beliefs are often shaped by cognitive biases and laziness or lack of time force us to access the click-baits or to share unscientific revelations. Machines are more patient, and if programmed carefully, could avoid such errors by fastidious analysis, not only the sources but also confirm contents via throughly scanned newspapers, research papers, history books. But the machine reading field is not there yet, so for time being we have to test easier solutions and use surface methods as identifying and classifying the source, analyzing the appearance of a page or writing style of its creator (Akamine et al. 2010).

Moreover, few last years have showed another problem with Big Data and machine learning, i.e. artificial intelligence systems acquiring stereotypes associated so far only with human beings (Bolukbasi et al. 2016; Caliskan, Bryson, and Narayanan 2017). Not dealing with this problem might end with a dialog system stating that woman's place is in the kitchen, all grandmothers are white (knowledge form any image search engine), and items recommended by people with non-Western names will be less trustworthy. There are several methods for unbiasing the data, abstracting or altering concepts is two possible option we consider. Instead of man or woman, "a person" can be used, although the data would need a few layers of semantical specificity because the knowledge of gender is often important for understanding. Removing bias manually from the data is laborious, and we believe that automatic discovery of reasons behind the stereotypes would be an ideal scenario. Removing any problematic concept from knowledge could lead to false discoveries, therefore several experiments must be performed to see if the oracle is able to find enough examples of stereotypes.

[4] Until these technologies are reliable and the annotated data widely accessible.

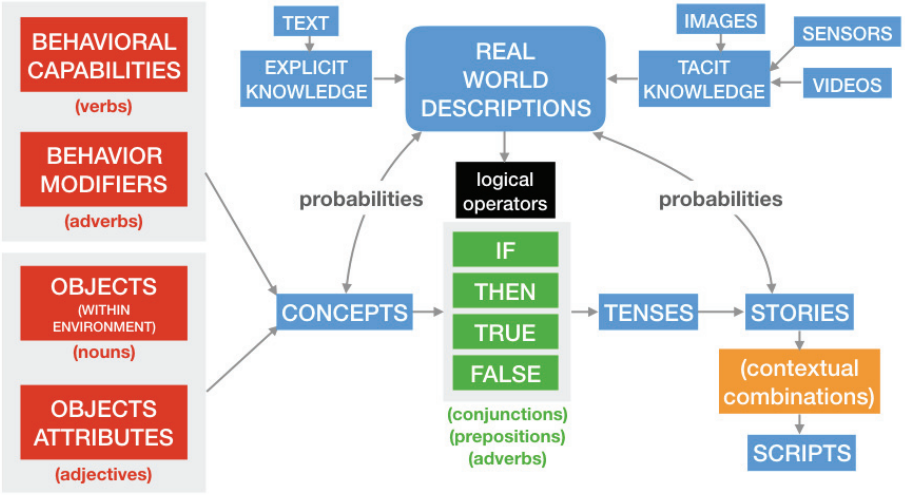


Figure 2: From words, through concepts, to stories. Basic idea of adding tacit knowledge by forcing linguistic descriptions and using probabilities of concept combination to induce possible and usual situations.

Philosophical Stance (or Lack of It)

Our experience with both robotic and non-robotic systems suggest that not only embodiment is unnecessary for moral decisions but also there is probably no need for subjectivity connected to consciousness often declared as the foundation of human ethical domain (Nath and Sahu 2017). Because our approach is rather pragmatic in its nature and we usually give rather scarce explanations about the bigger picture, we decided to use this section to explain some points which are very often misunderstood by our critics.

Provoking Philosophy by Avoiding It

Principally, we want to avoid adhering to any particular ethical school of thinking, although example-based approach might be used for testing utilitarian (by calculating utilities) and deontological (by extrapolating rules) systems. There are some ideas in modern of ethics which can be easily attached to our strategy, for instance an idealized ethical advisor is discussed by various philosophers (Sidgwick 1907; Firth 1952; Rawls 1971; Harsanyi 1977). (Sobel 1994) and (Rosati 1995) are probably the main critics of such all-knowing moral agent and the former describes four objections which our system could be referred to. The first one suggests that an ideal advisor could get lost in too many, always changing perspectives. As we show in Figure 4 always growing knowledge is not the obstacle but the opposite. Controlling timeline (as the consequences change with history) should be performed to avoid discovering polarities which were different a century ago, e.g. reactions to public lynches. Sobel's second and third objections applies to agent's experience: evaluation of one life can be evaluated only if it is experienced and this experience biases the agent when experiencing another one. Similar argument can be made about artificial agent which is given one set of experiences but in our case maximal number of experiences is used and forgetting one to process another is not necessary. The last objection argues that the Ideal Agent with perfect knowledge can conclude that non-perfect agents' is not worth living due to its limitations. To make robot with our system implemented kill anyone, the vast majority of stories would need to contain examples that killing is good, which is not true (the scale of actual data is shown in Figure 4). The same can be said about any utility maximizer often shown as an exemplification of dangerous AI. By changing the focus from theory to experience we our approach is closer to what Johnson calls "moral imagination". In (Johnson 1994), he challenges traditional ethics by emphasizing the role of stories we are confronted with from very early stage of our lives. Equipped with empathy we process examples from children's books, novels, movies. Our morals keep evolving as we are experiencing stories in our own lives, both by observing them and taking an active part.

Addressing Risks and Limitations of Machine Ethics

The complicated character of human ethics raises questions about risks and limitations of processing moral problems by non-human agents. (Brundage 2014) lists problems of the emerging field suggesting the whole endeavor might be pointless. As our systems need moral decision as we speak, we disagree with the main line of the critique, but agree with some points and believe they should be addressed. The problem of insufficient knowledge, complexity and/or the possible lack of computational resources is what we plan to solve by constructing a vast contextual ontology which should grow with the progress of both knowledge acquisition and computational capacities of hardware. Brundage points out that machine ethics is not able to make (or not to make) an exception to a rule when an exception shouldn't have been made based on the morally relevant factors. As our approach does not rely on any hard-coded rules and is supposed to discover and analyze as many factors as possi-

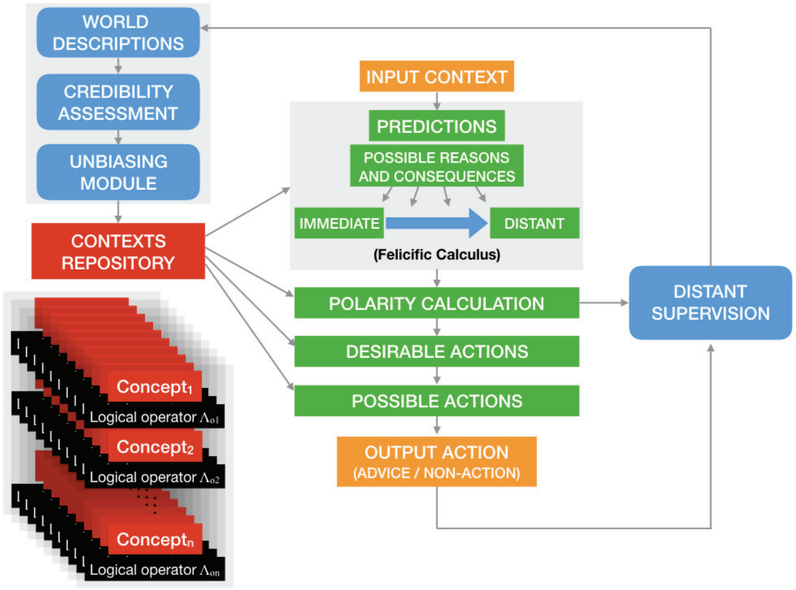


Figure 3: Unbiased collective intelligence as a source for machine learning: by giving the system examples of human experience could lead to richer reasoning about reasons and consequences of human / robot acts.

ble, dealing with exceptions should be easier than in other approaches. It is difficult to ensure perfect decision because there always might be a better one, but with unbiased knowledge and analytical power, a machine (at least in theory) might be a better and faster judge than average human being. Another set of possible problems is related to moral dilemmas facing an agent when it needs to sacrifice something important. Contextual knowledge based on real stories with reasons and consequences should contain examples of sacrifices which makes the problem of insufficient data most important to deal with. So called “folk morality” is often flawed, as Brundage notices. For that reason we concentrate on observing consequences, not on how people reason. He also worries that extrapolation of our values may be far beyond our current preferences. In our opinion, restricting our algorithm with common sense boundaries should prevent AI from becoming too creative and stop aligning with our values.

Conclusions and Future Work

Various models of moral judgement have been proposed and can be used in Artificial Moral Agents development, for example (Dehghani et al. 2008; 2008; Nado, Kelly, and Stich 2009; Ord 2015). On the other hand, empirical methods slowly enter the field of ethics and show, among others, how morals differ between cultures (Buchtel et al. 2015) or that *feeling right* is often more important than *feeling good* (Tamir et al. 2017). With this paper emphasizing the importance of the empirical (observational) side of ethical reasoning, we would like to spark a discussion about collecting, storing and normalizing contextual knowledge (chains of very specific concepts instead of very general single concepts). We believe that such knowledge could be very helpful in extrapolating rules, learning possible outcomes or testing existing systems. We believe that natural language, even being fuzzy and incomplete, can be a safe interlayer between the real world and abstract notions like ethics.

As computer scientists we often tend to model the world in a strict manner, we prefer to control input and output so the proposed algorithms can be easily tested and the results be published. But the value alignment may require us to share a significant part of the control to the world around us (by descriptions of it). For six million years we have gathered knowledge which becomes more and more accessible for machines and we believe it would not be smart if we ignore the contextual variety of “good vs. bad” stories humankind keeps accumulating. We believe that taming this knowledge may accelerate the progress of safe AI on a larger scale that is usually seen. It might be easier and faster to program a machine to acquire logics by analyzing moral cases than program logics to acquire morality. Whichever method will be most robust and “just”, the knowledge will be their common ground.

There are various approaches how to define the inborn instincts of justice. But a computer could learn from manifestations of those instincts without understanding them. As computer pattern recognition capabilities constantly grow, AI climbs bastions of human intelligence one after another. As Watson was more often correct than the best humans, some AMA can be more often “right” than all of us. Without any thinking, consciousness, free will but massive (multicultural) collective intelligence with decreased bias and increased credibility might be helpful not only to AI systems but also to anyone of us, even if in a form of mere voice

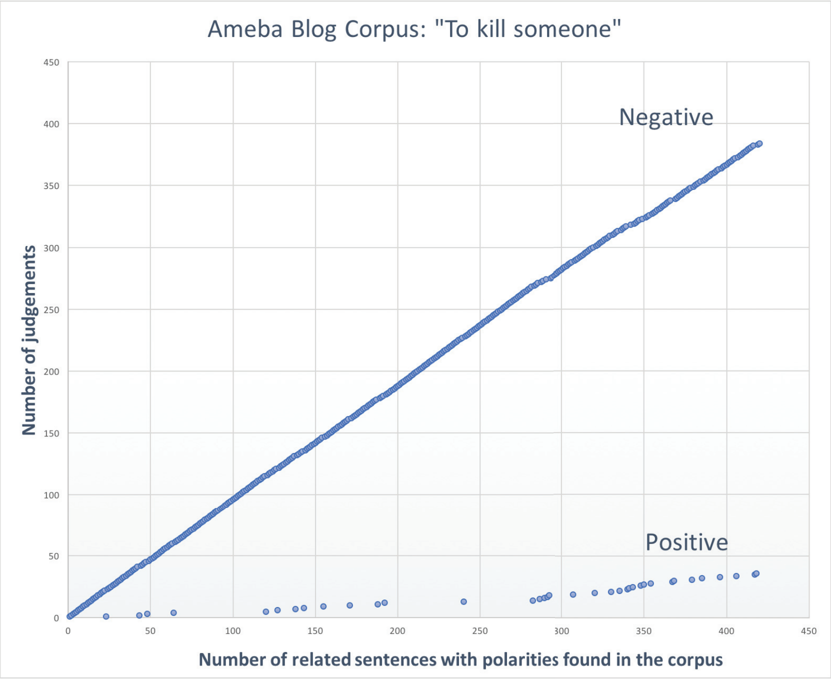


Figure 4: Importance of experience data size: Although number of positively labelled sentences about killing somebody also increases with new examples, the increase of correct (negative) consequence estimation is significantly higher.

assistant.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number 17K00295. We thank the Centre of Excellence for the Dynamics of Language at Australian National University for providing an ideal environment for transdisciplinary discussion that lead to crystalizing ideas described in this paper. We would also like to thank Dr. Siva Kalyan for enriching our background knowledge.

References

Akamine, S.; Kawahara, D.; Kato, Y.; Nakagawa, T.; Leon-Suematsu, Y. I.; Kawada, T.; Inui, K.; Kurohashi, S.; and Kidawara, Y. 2010. Organizing information on the web to support user judgments on information credibility. In *Universal Communication Symposium (IUCS), 2010 4th International*, 123–130. IEEE.

Anderson, M.; Anderson, S. L.; and Armen, C. 2006. MedEthEx: A prototype medical ethics advisor. In *Proceedings of the 18th Conference on Innovative Applications of Artificial Intelligence - Volume 2*, IAAI'06, 1759–1765. AAAI Press.

Arnold, T., and Scheutz, M. 2016. Against the moral turing test: accountable design and the moral reasoning of autonomous systems. *Ethics and Information Technology* 18(2):103–115.

Bolukbasi, T.; Chang, K.-W.; Zou, J. Y.; Saligrama, V.; and Kalai, A. T. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, 4349–4357.

Bostrom, N. 2014. *Superintelligence: Paths, dangers, strategies*. OUP Oxford.

Bretz, S., and Sun, R. 2017. Two models of moral judgment. *Cognitive Science* n/a–n/a.

Bringsjord, S.; Arkoudas, K.; and Bello, P. 2006. Toward a general logicist methodology for engineering ethically correct robots. *IEEE Intelligent Systems* 21(4):38–44.

Brundage, M. 2014. Limitations and risks of machine ethics. *Journal of Experimental & Theoretical Artificial Intelligence* 26(3):355–372.

Buchtel, E. E.; Guan, Y.; Peng, Q.; Su, Y.; Sang, B.; Chen, S. X.; and Bond, M. H. 2015. Immorality east and west: Are immoral behaviors especially harmful, or especially uncivilized? *Personality and Social Psychology Bulletin* 41(10):1382–1394.

Caliskan, A.; Bryson, J. J.; and Narayanan, A. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science* 356(6334):183–186.

Conitzer, V.; Sinnott-Armstrong, W.; Borg, J. S.; Deng, Y.; and Kramer, M. 2017. Moral decision making frameworks for artificial intelligence. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17) Senior Member / Blue Sky Track*.

Core, M. G.; Lane, H. C.; Van Lent, M.; Gomboc, D.; Solomon, S.; and Rosenberg, M. 2006. Building explainable artificial intelligence systems. In *AAAI*, 1766–1773.

Dehghani, M.; Tomai, E.; Forbus, K. D.; and Klenk, M. 2008. An integrated reasoning approach to moral decision-making. In *AAAI*, 1280–1286.

Firth, R. 1952. Ethical absolutism and the ideal observer. *Philosophy and Phenomenological Research* 12(3):317–345.

Hadfield-Menell, D.; Russell, S. J.; Abbeel, P.; and Dragan, A. 2016. Cooperative inverse reinforcement learning. In *Advances in neural information processing systems*, 3909–3917.

Harsanyi, J. C. 1977. Morality and the theory of rational behavior. *Social research* 623–656.

Johnson, M. 1994. *Moral imagination: Implications of cognitive science for ethics*. University of Chicago Press.

Krawczyk, M.; Rzepka, R.; and Araki, K. 2016. Extracting location and creator-related information from wikipedia-based information-rich taxonomy for conceptnet expansion. *Knowledge-Based Systems* 108:125–131.

Lenat, D. B., and Guha, R. V. 1989. *Building Large Knowledge-Based Systems; Representation and Inference in the Cyc Project*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1st edition.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, 1003–1011. Association for Computational Linguistics.

Nado, J.; Kelly, D.; and Stich, S. 2009. Moral judgment. *The Routledge companion to philosophy of psychology. Routledge, Milton Park*.

Nakamura, T.; Rzepka, R.; Araki, K.; and Inui, K. 2017. Be more eloquent, professor ELIZA – comparison of utterance generation methods for artificial second language tutor. In *Proceedings of Linguistic and Cognitive Approaches to Dialog Agents (LaCATODA 2017) IJCAI 2017 Workshop, CEUR*, volume Vol-1926, 34–41.

Nath, R., and Sahu, V. 2017. The problem of machine ethics in artificial intelligence. *AI & SOCIETY* 1–9.

Ng, A. Y.; Russell, S. J.; et al. 2000. Algorithms for inverse reinforcement learning. In *Icml*, 663–670.

Ord, T. 2015. Moral trade. *Ethics* 126(1):118–138.

Pereira, L. M., and Saptawijaya, A. 2016. *Programming machine ethics*, volume 26. Springer.

Rawls, J. 1971. A theory of social justice. *Cambridge, MA: Belknap*.

Rosati, C. S. 1995. Persons, perspectives, and full information accounts of the good. *Ethics* 105(2):296–325.

Rzepka, R., and Araki, K. 2005. What statistics could do for ethics? - The idea of common sense processing based safety valve. *AAAI Fall Symposium on Machine Ethics, Technical Report FS-05-06* 85–87.

Rzepka, R., and Araki, K. 2012. Polarization of consequence expressions for an automatic ethical judgment based on moral stages theory. Technical report, IPSJ.

Rzepka, R., and Araki, K. 2015. *Rethinking Machine Ethics in the Age of Ubiquitous Technology*. Hershey: IGI Global. chapter Semantic Analysis of Bloggers Experiences as a Knowledge Source of Average Human Morality, 73–95.

Rzepka, R., and Araki, K. 2017. What people say? Web-based casuistry for artificial morality experiments. In Everitt, T.; Goertzel, B.; and Potapov, A., eds., *Artificial General Intelligence - 10th International Conference, AGI 2017, Melbourne, VIC, Australia, August 15-18, 2017, Proceedings*, volume 10414 of *Lecture Notes in Computer Science*, 178–187. Springer.

Rzepka, R.; Mitsuhashi, K.; and Araki, K. 2016. Avoiding green and colorless ideas: Text-based color-related knowledge acquisition for better image understanding. In *Proceedings of the 4th International Workshop on Artificial Intelligence and Cognition co-located with the Joint Multi-Conference on Human-Level Artificial Intelligence (HLAI 2016), CEUR Vol-1895*, 38–44.

Schank, R., and Abelson, R. 1977. *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*. Hillsdale, NJ.: Lawrence Erlbaum Associates.

Shudo, S.; Rzepka, R.; and Araki, K. 2016. Automatic evaluation of commonsense knowledge for refining japanese conceptnet. In *The 12th Workshop on Asian Language Resources*, 105.

Sidgwick, H. 1907. *The methods of ethics*. Hackett Publishing.

Sobel, D. 1994. Full information accounts of well-being. *Ethics* 104(4):784–810.

Speer, R., and Havasi, C. 2012. Representing general relational knowledge in conceptnet 5. In Chair), N. C. C.; Choukri, K.; Declerck, T.; Doğan, M. U.; Maegaard, B.; Mariani, J.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. Istanbul, Turkey: European Language Resources Association (ELRA).

Takagi, K.; Rzepka, R.; and Araki, K. 2011. Just keep tweeting, dear: Web-mining methods for helping a social robot understand user needs. In *Proceedings of Help Me Help You: Bridging the Gaps in Human-Agent Collaboration*, 60–65. Symposium of AAAI 2011 Spring Symposia (SS-11-05).

Tamir, M.; Schwartz, S. H.; Oishi, S.; and Kim, M. Y. 2017. The secret to happiness: Feeling good or feeling right? *Journal of Experimental Psychology: General* 146(10):1448.

Tenenbaum, J. B.; Kemp, C.; Griffiths, T. L.; and Goodman, N. D. 2011. How to grow a mind: Statistics, structure, and abstraction. *science* 331(6022):1279–1285.

Trappl, R. 2015. *A Construction Manual for Robots' Ethical Systems*. Springer.

Vinyals, O.; Toshev, A.; Bengio, S.; and Erhan, D. 2015. Show and tell: A neural image caption generator. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Towards Provably Moral AI Agents in Bottom-Up Learning Frameworks

Nolan P. Shaw, Andreas Stöckel, Ryan W. Orr,
Thomas F. Lidbetter, Robin Cohen
David R. Cheriton School of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
{nolan.shaw, astoecke, rworr, finn.lidbetter, rcohen}@uwaterloo.ca

Abstract

We examine moral decision making in autonomous systems as inspired by a central question posed by Rossi with respect to moral preferences: can AI systems based on statistical machine learning (which do not provide a natural way to explain or justify their decisions) be used for embedding morality into a machine in a way that allows us to prove that nothing morally wrong will happen? We argue for an evaluation which is held to the same standards as a human agent, removing the demand that ethical behavior is always achieved. We introduce four key meta-qualities desired for our moral standards, and then proceed to clarify how we can prove that an agent will correctly learn to perform moral actions given a set of samples within certain error bounds. Our group-dynamic approach enables us to demonstrate that the learned models converge to a common function to achieve stability. We further explain a valuable intrinsic consistency check made possible through the derivation of logical statements from the machine learning model. In all, this work proposes an approach for building ethical AI systems, coming from the perspective of artificial intelligence research, and sheds important light on understanding how much learning is required in order for an intelligent agent to behave morally with negligible error.

1 Introduction

In her 2016 article "Moral Preferences" (Rossi 2016), Francesca Rossi raises the question of how morality could be embedded into machines. Considering ongoing automation, the growing autonomy of AI systems, and their deployment in safety-critical applications, it becomes increasingly urgent to find answers to this question. Rossi suggests seven largely independent research directions which help to shed light on the larger issue. One of these questions concerns the correctness of moral decisions learned with statistical approaches, such as neural networks, under the prior assumption that moral decisions can be formalized in this way. Since it is arguably hard to inspect the inner workings of a trained statistical learning model, ensuring that the model behaves as intended—even in situations not anticipated by its creators—is of particular importance.

The argument we present here is threefold. First, proving anything about morality in a wholly objective fashion is impossible[1], since morals emerge from societies and are only meaningful in the group context that gives rise to them (Section 2). In other words, while we can identify desirable meta-characteristics of a moral system (Section 3), the same cannot be said for capturing the moral rules themselves. Second, even if we were to ignore our first point and assume that we are able to derive arbitrary amounts of training data, making sure that a statistical learning system has a small generalization error is difficult. The model that is being trained to perform actions must be specifically tailored to the problem at hand and given large quantities of training data (Section 4). Third, we propose a group-dynamic (multi-agent feedback) approach as an alternative to ensuring that the trained model behaves morally. Since we should not subject machines to higher standards than humans, it suffices to show that the learned morals converge to a common decision function (Section 5). We further argue that it should be possible to derive logical statements from the machine learning model, providing machines with an intrinsic consistency check (Section 6). We conclude with a proposed system architecture for a group of autonomous agents.

Bottom-up learning methods such as deep neural networks will likely be a crucial component in future AI systems, including those obliged to render morally relevant decisions. While trained statistical models are reputed to be difficult to analyze in terms of an underlying decision process, in this paper we aim to demonstrate that they may still be suitable for morally relevant tasks.

2 Limits of Provability

Before we can address the principal question of how we can prove that an agent will act morally, we must first recognize that any attempt at answering this question will face limitations. A complete solution would require some objective notion of morality: some measure by which any action could be judged as either moral or immoral in a general setting. However, current theories of ethics make this an impossible task, simply because the morality of an action is dependent on the ethical framework in which it is judged. For instance, there are many imaginable scenarios where Immanuel Kant's deontological ethics theory is at odds with John Stuart Mill's

[1] What we mean to say is that there are no fine-grained moral laws, not that there is no objectivity in moral laws whatsoever.

utilitarian ethics. Hence, there can be no blanket solution to the problem. The morality of an action can only be proved with respect to some particular ethical theory, if at all.

Of course, there are many possible situations where well-established ethical frameworks will be in agreement. In such cases one could argue that there is an objectively moral decision that is not specific to any particular framework. However, due to the complexities and intricacies of the various ethical frameworks, scenarios where these theories are all in agreement may be highly constrained. Conflicting judgments of morality begin to arise more often once the contexts in which decisions are made become too general. It is then the generality of the application which prohibits provably moral decisions, with respect to multiple theories of ethics. One may only be able to prove results on the morality of an agent's actions if the environment in which it is making decisions is sufficiently constrained, and the moral framework is specified. Hence, to be able to prove desirable properties of our moral agent independent of any framework we will establish a set of guiding meta-moral qualities and assume a constrained application. The nature of this constrained learning is discussed in Section 4. A proposed solution to evaluating moral behaviour more generally will be the topic of Section 5.

3 Standards Demanded of a Moral Agent

First, we consider the standards that a machine must meet in order to be a proper moral agent. If it is required that the machine be perfectly comprehensible and that we can ensure that it does no wrong before introducing it into society, then this task is infeasible. Meeting this requirement would demand that we can deterministically predict not only this agent's set of learned moral principles, but also the external conditions that would inform how it applies these principles to the myriad of moral decisions it would be faced with.

Instead, we first make a precise statement of exactly what benchmarks a machine ought to meet to be considered a moral agent. Currently, humans only have themselves as examples of autonomous moral agents. As such, we hold that a machine should not be required to meet any standards that humans may not meet themselves. This stipulation removes the need to prove the *means* by which an agent learns moral principles or behavior, focusing solely on the behavior and moral rules themselves. Furthermore, it removes the constraint of being able to prove that a machine will *never* do any wrong, as we do not hold humans to the same standard. Similar to the argument for self-driving cars, it is unimportant that machines be morally infallible (if this were even possible)—only that they do at least as well as humans. In addition, this stipulation ignores the demand that artificial agents behave in an acceptably moral manner until being provided with sufficient time to properly learn the moral values of its society. Finally, if we hold machines to the same standards as humans, then it is not the case that every machine converge to behavior that is ideal for its community, only that a population of such machines would largely abide by the moral laws of their society.

Next, we define a short list of meta-moral qualities that we demand machines possess, in order to be considered proper moral agents. This list is by no means meant to be exhaustive—rather it is meant to be as sparse as possible—but should certainly include:

1. **Robustness:** whatever moral architecture is developed must allow a machine to change its moral principles. What is considered 'good' may differ from community to community or over time. As such, artificial moral architecture must be adaptive. It is desired that an agent expresses this quality in two ways. First, it is desired that an untrained agent be able to adopt the moral laws of any society. Second, a trained agent should be able to eventually adopt new principles when transplanted for one society to another. This allows a machine to behave in a way that is relevant to its cultural environment.
2. **Consistency:** we hold that, regardless of what moral principles a machine learns, these principles are at least internally consistent.
3. **Universality:** taking a page from Kant's book, we hold that a machine's learned moral principles be universally applicable to all members of its society.
4. **Simplicity:** note that there is a concern with the combination of the above qualities: it is possible that a moral agent develop an extensive list of moral principles—all of which are consistent and may be universally implemented—yet overly restrictive and arbitrary. This stands in conflict with the first quality, and would make it plausible for a community of agents to sacrifice diversity for the sake of homogeneity (a quality we know to be undesirable for productivity and progress). As such, we make the additional assertion that a machine should always endeavor to operate on the smallest number of "firm" moral principles possible.

These qualities allow the moral agent to, at once, adopt a subjective set of principles that are relevant to the particular society it inhabits, while also ensuring that the moral agent has some objective ground upon which it can internally evaluate the strength of its principles independently of society.

4 Sufficient Conditions for a Provably Moral Behaviour

Having laid out the meta-qualities that we wish an agent to have, and holding that there is no objective measure for particular moral laws, we now turn back to the original question posed by Francesca Rossi. Can we, at least in theory, prove that an agent will correctly learn to perform moral actions given a set of samples within certain error boundaries? The answer is yes: assuming that we can generate an arbitrary amount of training samples in order to learn what actions to take, machine learning theory hands us sufficient conditions under which such a function can be learned with small error.

From a theoretical perspective, the process of acquiring moral behavior (i.e. learning moral principles) within a statistical learning framework can be formalized as approximating a function $f : C \to A$, where C is the set of possible moral contexts and A is the set of actions available to the agent. As a somewhat contrived example, consider an epidemic, where C describes properties of a disease (e.g., mor-

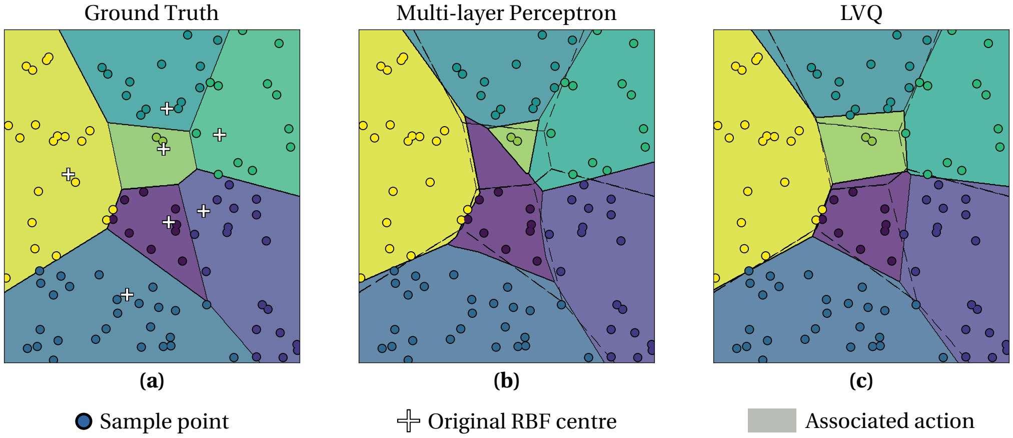


Figure 1: Importance of model selection. (a) depicts the ground truth $f(c)$ oblivious to the learner. Colored regions represent actions a_i. The underlying functions f_i are radial basis functions (RBFs) centered at the white crosses, over which f is the $\arg\max$, resulting in a Voronoi diagram. Colored circles correspond to training samples. (b) shows f as learned by a multi-layer perceptron. Dashed contour lines correspond to the ground truth. The function in (c) is learned with a variant of the learning vector quantization (LVQ) algorithm (Kohonen 1995), where the underlying assumption that the actions are assigned as nearest neighbors to prototypes results in a smaller generalization error (e.g., compare the center dark violet region in (b).

tality rate and contagiousness), and A is a set of actions such as administering an unsafe vaccine or isolating patients, each with their own merits, costs, and dangers. An optimal strategy would, depending on the context, perform the action which minimizes the number of deaths.

In an offline-learning scenario, the agent receives a set of samples $S \subset C \times A$ describing morally optimal behavior, with the goal to minimize the training error between a learned $\hat{f}$ and the set of samples. For finite actions $A = \{a_1, \ldots, a_n\}$ this process can be modeled as a multi-class learning problem, which—among other methods—can be solved by learning n separate functions, where individual $f_i : C \to \mathbb{R}$ correspond to the utility of action i in the given moral context. The function $\hat{f}$ selects the best among the learned actions, i.e. $\hat{f}(c) = a_j$, where $j = \arg\max_i \hat{f}_i(c)$.

To ensure morally optimal behavior, the learned $\hat{f}_i$ must have a small generalization error. As a direct result of the first *no free lunch* (NFL) theorem (Wolpert and Macready 1997), a small generalization error can only be guaranteed if the hypothesis space $\mathcal{H}$ containing the optimal f is specialized (Ho and Pepyne 2002). The NFL is formalized as

$$\sum_{f\in\mathcal{H}} P\big(d^y_m \mid f, m, a_1\big) = \sum_{f\in\mathcal{H}} P\big(d^y_m \mid f, m, a_2\big)\,, \tag{1}$$

where d^y_m is a sorted set containing the error for each training sample y, m is the number of training samples and a_1, a_2 are static learning algorithms subject to sensibility constraints laid out in (Wolpert and Macready 1997). Correspondingly, if all f in the hypothesis space are equally likely to be the "true" ground truth, a learning algorithm which performs particularly well on a subset $\mathcal{H}_1 \subset \mathcal{H}$ must, on average, perform worse for the remaining $\mathcal{H}$ for eq. (1) to hold.

For example, if we have prior knowledge that f resides in a hypothesis space $\mathcal{H}$ produced by a parametric mathematical model, we can expect to fit the model parameters to our data with relatively small generalization error. On the other hand, for unconstrained $\mathcal{H}$—that is, the set of all possible functions mapping from C to A—we cannot, on average, expect to perform better than a function in that space found by a random optimizer. While the NFL theorem seems counter-intuitive given recent advances of machine learning approaches, the effectiveness of neural networks and back propagation can potentially be explained as an implicit restriction of $\mathcal{H}$ to a set of "naturally occurring" functions (Lin, Tegmark, and Rolnick 2017). In the context of learning moral actions, these implicit restrictions are far too vague to make any guarantees.

So, moral actions, for which a small generalization error is crucial (cf. section 3), can only be learned in the framework presented above if we assume that the "true" strategy is part of a well-assessable function family for which a matching machine learning algorithm exists. The example depicted in fig. 1 illustrates this: while both algorithms classify the training samples with zero error, the more constrained model and learning algorithm result in a significantly reduced generalization error.

Assuming that we are able to develop a model and the corresponding hypothesis space, $\mathcal{H}$, we may ask how many samples have to be (uniformly) sampled from the input space C to guarantee a certain maximum generalization error ε. Here, machine learning theory provides the concept of *probably approximately correct* (PAC) learning (Valiant 1984). For a discretized hypothesis space of size $|\mathcal{H}|$, a maximum error ε, and success probability $1 - \delta$, a lower bound for the required sample count m is given as (Shalev-Shwartz and Ben-David 2014)

$$m \geq \frac{1}{\varepsilon}\Big(\ln\big(|H|\big) - \ln\big(\delta\big)\Big)\,. \tag{2}$$

Essentially, for a model with d parameters, and k dis-

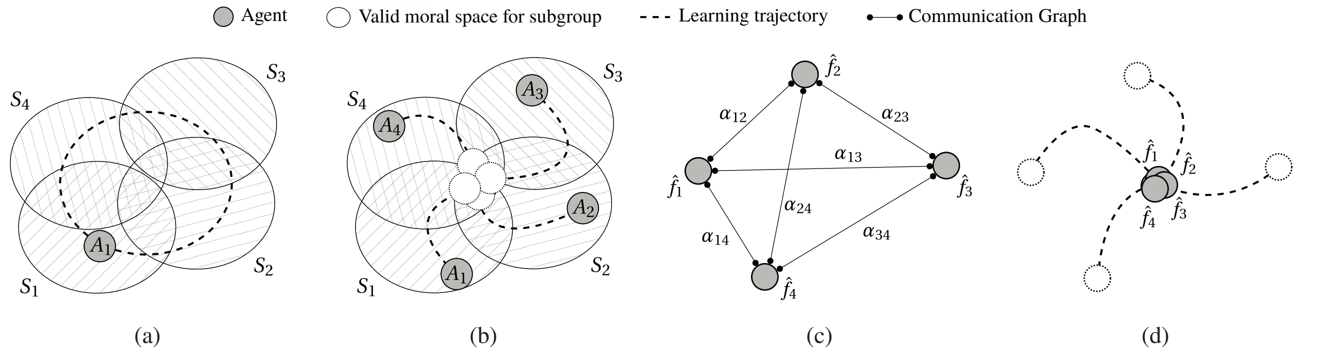


Figure 2: Networks of agents. Development of a learned moral decision function $\hat{f}$ in a single- and multi-agent environment (a, b) while transitioning through multiple subgroups. If the communication graph of a multi-agent system is connected (c), the value represented by the agents—here a learned moral decision function $\hat{f}$—will converge to a single point (d).

cretization steps per parameter, m is linear in d, since $\mathcal{O}\big(\ln\big(k^d\big)\big) = \mathcal{O}\big(d\big)$. In practice far fewer samples may be required; however, no guarantees can be made other than those in eq. (2) without more specific information about $\mathcal{H}$.

While the above theories provide a set of sufficient constraints for the problem at hand, finding a consistent model and acquiring a large set of training samples may prove to be harder than the problem that machine learning aims at solving in the first place—namely having to explicitly model top-down moral decisions. Yet, not all hope is lost: even if a learning framework does not strictly fulfill the above criteria, we next propose strategies for evaluating a learned moral function in the context of multi agent systems and the consistency of learned moral rules.

5 Proving Stability: Analyzing Networks of Agents

A single learning agent can satisfy some of the qualities outlined in Section 2 by itself; it can be designed with a learning algorithm sufficiently robust to adapt to a new set of moral principles, it may internally check its moral principles to ensure consistency, and it can be designed to search for the simplest set of morals possible by itself. However, a single agent may struggle with the meta-quality of universality. For example, an agent deployed to a society with multiple subgroups may continually adapt to each individual subgroup, rather than properly generalizing to the set of morals encompassing the complete society, as shown in fig. 2a. Typically, this problem would be solved by gradually decreasing the learning rate of the agent, but such an approach would remove the agent's ability to generalize to a new society, violating our standard of robustness. Instead, we propose using a multiagent system to explore the moral space from multiple perspectives simultaneously, and require that the agents eventually converge to a stable set of moral principles (fig. 2b). The agents in the system will essentially operate under Kant's categorical imperative: "Act only in accordance with that maxim through which you can at the same time will that it become a universal law" (Kant 1993).

In the multiagent model, each agent would be designed as a learner which accepts context-action pairs (c_l, a_l) as input, and learns a set of moral principles such that the agent is capable of selecting a morally acceptable action a_m when presented with a context c_m. By learning from the provided samples of human morality, each agent will individually learn a set of moral principles. Convergence of multiple agents to a single set of moral principles is then similar to the consensus problem in coordinating multiagent networks. For a continuous-time system, the solution to the consensus problem is defined as (Ren, Beard, and Atkins 2005)

$$\dot{x}_i = -\sum_{j \in J_i(t)} \alpha_{ij}(t)(x_i(t) - x_j(t)). \tag{3}$$

This algorithm essentially works as a weighted average of all agents in the system, as an agent i compares its current value, $x_i(t)$ to each value represented by all connected agents, $x_j(t), j \in J_i(t)$, where $J_i(t)$ is the set of all other agents currently connected to agent i. If the moral space the system is exploring is able to be modeled in such a way where the derivative and difference operators can be defined, this equation is directly applicable to the multiagent system. For cases where those operators cannot easily be defined, the learning algorithm can be adopted to mirror this equation. Each time an agent i takes an action a_i, it would broadcast the context-action pair (c_i, a_i) to all other connected agents in the set $J_i(t)$. Each connected agent j can then use the (c_i, a_i) pair as a new sample point for learning, and adapt its morals to be similar to agent i. In addition to fulfilling our desired property of universality, the solution to the consensus problem described by fig. 3 results in a provably stable consensus in a multiagent system. As long as the agents are in contact with each other frequently enough[2], convergence is guaranteed (Ren, Beard, and Atkins 2005), as shown in figs. 2c, 2d.

However, complete consensus in a multiagent system may not be desirable. For example, there could be two subgroups in society with disjoint moral principles, and a full consensus across all agents would lead to a set of morals which does

[2] Where communication does not occur for longer periods, we arrive at the "multi-society" case discussed in the next paragraph.

not properly satisfy the needs of either subgroup. To address this problem, inspiration can be taken from how humans develop differing morals. Humans learn morality by observing and learning from the moral actions of others, but we do not take an average of all observed actions. Instead, we model a level of trust in other humans, and use that level of trust to determine how to learn from another person's actions (Hahn 2017). By determining which actions to learn from, humans can form separate sets of moral principles specialized to specific contexts. Trust modeling can be adapted to a moral multiagent system in a similar manner, to allow specialization for different societies. Simulations have shown that using a Bayesian model of trust can result in either agreeing clusters or polarized disagreeing clusters when modeling the validity of information received from other agents (Olsson 2013). In the context of a moral multiagent system, agents could attempt to model the probability that other agents in the system are attempting to follow the same set of moral principles as themselves. Agents can use a basic Bayesian calculation to model this probability,

$$P(M \mid a) = \frac{P(a \mid M)P(M)}{P(a \mid M)P(M) + P(a \mid \neg M)P(\neg M)}, \quad (4)$$

where M is the event that an observed agent is acting morally (at least according to the observing agent's current moral principles), and a is an action taken by the observed agent. Using eq. (4), if agent i observed agent j taking action a_j, agent i would estimate if a_j is a valid moral action based on $x_i(t)$—i's current moral principles. If a_j is deemed moral by i, i can increase its trust in j, which would increase the consensus weighting parameter α_{ij} from eq. (3). Conversely, if a_j is deemed immoral, i can decrease its trust in j, reducing α_{ij}. In cases where j is deemed fully immoral relative to i, the α_{ij} parameter could be set to zero, causing i to ignore all of j's actions.

Using a Bayesian approach to model the possible morality of other agents in the system, agents would be allowed to form disagreements in their definitions of moral principles, while enforcing convergence to one or more clusters of agents via the α_{ij} parameter. Allowing multiple clusters increases the overall universality and robustness of the multiagent system, by ensuring any necessary morally specialized agents can be formed. Since artificial agents in the system are learning directly from humans (initially trained offline using human data), the system is expected to converge to a stable point within the space of human moral principles, while satisfying the meta-moral qualities desired of a moral agent.

Any agent which is able to learn from other agents can be used in this system and will achieve consensus with the other agents within a cluster, i.e. there is guaranteed convergence to a common moral preference function. The agent's ability to learn is the only property which governs whether this convergence will occur, whereas the communication frequency and trust models dictate which clusters will result from convergence. It is important to note that agents in the system may in fact be humans and not just artificial agents. Regardless, we would still expect convergence, since humans are also exposed to moral actions from which they can learn.

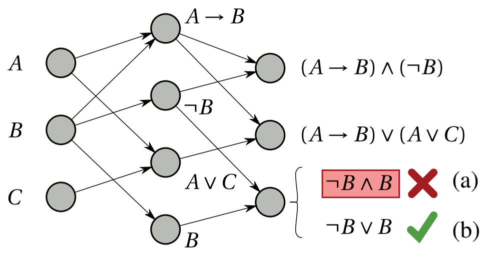


Figure 3: Illustration of the extraction of logical expressions from a neural network. Whereas there is no consistent variable assignment satisfying the expressions in (a), the network in (b) has a possible variable assignment ($a = 0, b = 0, c = 1$).

6 The Consistency of Learned Moral Rules

Another means of evaluating the learned principles an agent develops is to consider the consistency of the rules it learns. Assume, for instance, that we are working with a hierarchical learning system, such as a neural network. We can label the input layer (which corresponds to the morally relevant variables) as atomic formulas. From there, we may assign a logical sentence built out of these atoms that best fits each node and evaluate the internal consistency of these groups of sentences.[3] The result is a self-checking system that raises an internal red flag any time an inconsistency is found between the sentences of this network, at each layer of the neural network (fig. 3). If a red flag is raised, then the agent must change or discard one of its conflicting moral principles.

One concern with this approach is that it is not computationally feasible to constantly assign and evaluate all the sentences each time the weights in the network are updated, since such neural networks can be extremely large. However, the goal is not to guarantee that inconsistency never occurs. It is only to evaluate these networks as best as possible. Again, we turn to the standards that people meet as justification that this is sufficient for machine agents as well. It is infeasible to demand that a human moral agent be perfectly consistent in order to participate in society—only that they reevaluate their principles once an inconsistency is found. Figure 4 provides an overview of the architecture resulting from the above considerations. The agent is bootstrapped with a classically learned machine learning (ML) model, subject to the constraints laid out in Section 4. When deployed, the agent uses its model to make moral decisions. Observations of moral actions in the environment, triples (c, a, α), where α is the trust in the agent the action originated from, are integrated into an updated model. This updated model is constantly checked for logical consistency, and newly deduced moral rules are used to further enhance the model. Furthermore, in addition to sole interaction with the environment, we propose to run an internal multi-agent simulation akin to Section 5, to ensure that the moral rules

[3] Note that we do not propose a comprehensive analysis of the learned model. We instead extract individual logical statements, which is more feasible than the general problem of explaining the learned decision process.

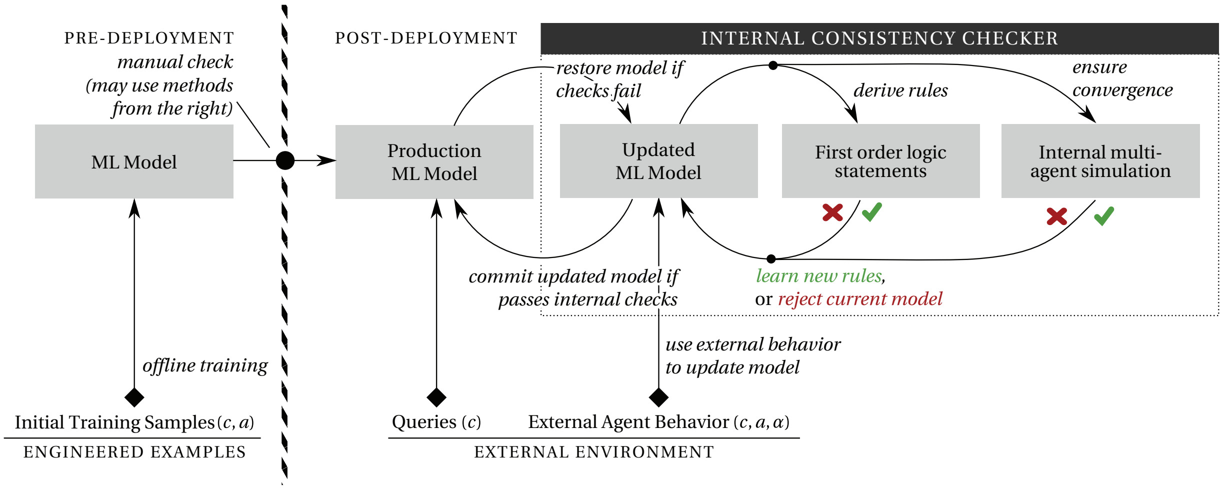


Figure 4: Proposed agent architecture. Left part of the diagram refers to an initial training phase, right part to the agent as it would be deployed in an actual environment. See text for description.

indeed lead to stability. Once the updated model passes these checks, it is swapped with the current model.

7 Discussion and Conclusion

To summarize, we hold that an artificial moral agent be held to the same standards as a human agent. We do not demand that such an agent justify the means by which it learns its moral principles, nor do we demand that an agent always act in a manner that society deems ethical. However, we do demand that any moral framework possesses the short list of meta-qualities we have outlined.

Acknowledging the limits of learning moral behavior, we may nevertheless prove how much learning is required in order for a moral agent to behave morally with negligible error. Furthermore, we may prove that an artificial moral agent can be expected to adopt human morals when introduced into a society of human agents, by using Bayesian models of trust to inform its moral decisions. In addition to being able to evaluate the moral behavior of an agent, we may also evaluate the moral principles an agent learns by evaluating their internal consistency.

Similar to other researchers, we have imagined a training phase in which agents may learn how to act ethically. Conitzer et al. (2017) also discuss moral decision-making frameworks where machine learning uses a set of moral decision problem instances labeled with human judgments. They comment on the challenge of identifying all the key features for the training. In our case, we have advocated adherence to four central properties as the basis for considering the actions as morally acceptable, though we also acknowledge the difficulties in identifying moral features with greater specificity. Other researchers have examined verifiably ethical behavior of agents. Dennis, Fisher, and Winfield (2015) focus on the case of robots and promote the value of model checking methods. Another paper related to our work is Armstrong (2015), which discusses the relative advantages of using predetermined ethical preferences, as opposed to enabling agents to learn values (including those from their environments). We believe that hard-coded values sacrifice robustness and run the risk of introducing human bias on the part of the developer. The learning-based approach has the advantage of being flexible, and Section 5 addresses the concern that an AI agent will not adopt human values when placed in a human society. The advantage of logical representations to enable ethical judgment by agents is also promoted in Cointe, Bonnet, and Boissier (2016); our work hopes to use these representations to construct the internal consistency checker outlined in Section 6. Anderson and Anderson (2015) suggest that a consensus of ethicists should determine what is morally acceptable for an agent's behavior. Provided that a framework uses the Bayesian models of trust outlined in Section 5, self-made decisions from agents should already align with society's values without the need for such a prescribed code of ethics.

We also propose a list of "next steps" for this research area. First, we must construct metrics for measuring various moral factors, so that a proper training set may be developed for learning. Second, a proof of concept must be developed for the online consistency checker proposed. Finally, it is our hope that once these first two implementation challenges are solved, we may build a multi-agent system to verify that convergence of moral behavior really does happen over time. Once these technical hurdles have been overcome, we will be much closer to artificial moral agents that not only act in accordance with human values, but are active participants in developing ethics in society.

References

Anderson, M., and Anderson, S. L. 2015. Toward ensuring ethical behavior from autonomous systems: a case-supported principle-based paradigm. *Industrial Robot: An International Journal* 42(4):324–331.

Armstrong, S. 2015. Motivated value selection for artificial agents. In *AAAI Workshop: AI and Ethics*.

Cointe, N.; Bonnet, G.; and Boissier, O. 2016. Ethical judgment of agents' behaviors in multi-agent systems. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 1106–1114. International Foundation for Autonomous Agents and Multiagent Systems.

Conitzer, V.; Sinnott-Armstrong, W.; Borg, J. S.; Deng, Y.; and Kramer, M. 2017. Moral decision making frameworks for artificial intelligence. In *AAAI*, 4831–4835.

Dennis, L. A.; Fisher, M.; and Winfield, A. F. 2015. Towards verifiably ethical robot behaviour. In *AAAI Workshop: AI and Ethics*.

Hahn, U. 2017. Rationality and the role of limited experience. Invited Talk, 39th Annual Conference of the Cognitive Science Society.

Ho, Y.-C., and Pepyne, D. L. 2002. Simple explanation of the no-free-lunch theorem and its implications. *Journal of optimization theory and applications* 115(3):549–570.

Kant, I. 1993. *Grounding for the metaphysics of morals: With on a supposed right to lie because of philanthropic concerns*. Hackett Publishing.

Kohonen, T. 1995. Learning vector quantization. In *Self-Organizing Maps*. Springer. 175–189.

Lin, H. W.; Tegmark, M.; and Rolnick, D. 2017. Why does deep and cheap learning work so well? *Journal of Statistical Physics* 168(6):1223–1247.

Olsson, E. J. 2013. A bayesian simulation model of group deliberation and polarization. In *Bayesian argumentation*. Springer. 113–133.

Ren, W.; Beard, R. W.; and Atkins, E. M. 2005. A survey of consensus problems in multi-agent coordination. In *American Control Conference, 2005. Proceedings of the 2005*, 1859–1864. IEEE.

Rossi, F. 2016. Moral preferences. In *The 10th Workshop on Advances in Preference Handling (MPREF), New York, NY, USA*.

Shalev-Shwartz, S., and Ben-David, S. 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Valiant, L. G. 1984. A theory of the learnable. *Communications of the ACM* 27(11):1134–1142.

Wolpert, D. H., and Macready, W. G. 1997. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation* 1(1):67–82.

Ethics as Aesthetic for Artificial General Intelligence

Dan Ventura
Computer Science Department
Brigham Young University
ventura@cs.byu.edu

Abstract

We address the question of how to build AI agents that behave ethically by appealing to a computational creativity framework in which output artifacts are agent behaviors and candidate behaviors are evaluated using a normative ethics as the aesthetic measure. We then appeal again to computational creativity to address the meta-level question of which normative ethics the system should employ as its aesthetic, where now output meta-artifacts are normative ethics and candidate ethics are evaluated using a meta-ethics-based aesthetic. We consider briefly some of the issues raised by such a proposal as well as how the hybrid base-meta-level system might be evaluated from three different perspectives: creative, behavioral and ethical.

Introduction

Artificial intelligence (AI) continues to mature and deliver on promises 50 years or more in the making, and this development has been especially marked in the last decade. However, as significant as these AI advances have become, the ultimate goal of artificial general intelligence is yet to be realized. Nevertheless, a great deal has been said about ethical issues arising from the development of AI systems (both the current specialized variety and the yet-quixotic general variety) that now can or may soon be able to impact humanity at unprecedented scale, with predictions ranging from the possible of a Utopian post-human immortality to the enslavement or even annihilation of the human race. Such discussions appear in every form imaginable, from monographs (Wallach and Allen 2008; Anderson and Leigh 2011; Müller 2016) to academic journals (Anderson and Anderson 2006; Muehlhauser and Helm 2012) to popular literature (Kurzweil 2005; McGee 2007; Fox 2009; Coeckelbergh 2014) to government studies (Lin, Bekey, and Abney 2008; European Parliament, Committee on Legal Affairs 2017). These treatments almost always take the form of applied ethics, either to be applied to humans doing the research that will inevitably lead to an AI-dominated future or to be applied to the AI systems themselves, or both. These discussions are most often normative in nature, though they can examine meta-ethics as well. Thus, we currently face the twin problems:

1. How can we ensure an AI agent behaves ethically?
2. What do we mean by ethical?

To begin with, we will simply postulate an abstract computational creativity (CC) approach for the implementation of an AI system. That is, we postulate a system whose domain of creation is behavioral policy, a system whose output artifacts are goals and/or decisions and/or sequences of actions. Given this admittedly ambitious premise and using a CC framework, we will argue the two questions can be naturally addressed. The question of how to impose an ethics on such a system can be addressed by implementing the CC system's aesthetic for evaluating artifacts as a (normative) ethics. In other words, that ethics acts as the filter by which the utility of system actions, decisions and goals is judged. The meta-level question of *which* normative ethics ought to be applied as the system's aesthetic can be addressed by allowing the system to create a suitable norm, given some meta-level aesthetic for ethics. That is, we suggest a CC system whose output artifact is a normative ethics and whose aesthetic is some way to evaluate said norm.

To summarize, we propose an appeal to computational creativity that answers both of our questions of interest:

1. We can build an ethical AI agent as a computational creativity system whose output artifacts are goals, decisions and behaviors and whose aesthetic component is a normative ethics.
2. We can delegate the choice of normative ethics to the AI agent by implementing a meta-level computational creativity system whose output artifacts are normative ethics and whose aesthetic is a meta-ethics.

Ethical Behavior Invention

The field of computational creativity has been described as "the philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviors that unbiased observers would deem to be creative" (Colton and Wiggins 2012). It has been characterized by attempts at building systems for meeting this standard in a wide variety of domains, including culinary recipes (Morris et al. 2012; Varshney et al. 2013), language constructs such as metaphor (Veale and Hao 2007) and neologism (Smith, Hintze, and Ventura 2014), visual

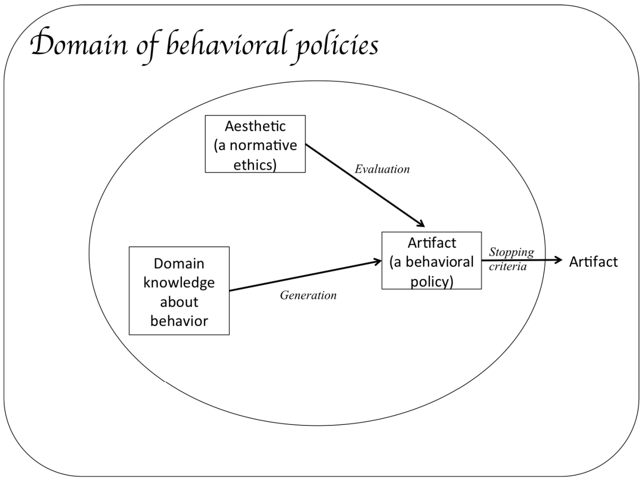

Domain of behavioral policies
Aesthetic (a normative ethics)
Evaluation
Domain knowledge about behavior
Generation
Artifact (a behavioral policy)
Stopping criteria
Artifact

Figure 1: A CC system embedded in the domain of behavioral policies uses domain knowledge about behavior to generate candidate policies that are vetted by an ethics-based aesthetic. Those polices judged to be of value by the aesthetic are exported to the domain, becoming viable policies for an AI agent.

art (Colton 2012; Norton, Heath, and Ventura 2013), poetry (Toivanen et al. 2012; Oliveira 2012; Veale 2013), humor (Binsted and Ritchie 1994; Stock and Strapparava 2003), advertising and slogans (Strapparava, Valitutti, and Stock 2007; Özbal, Pighin, and Strapparava 2013), narrative and story telling (Pérez y Pérez and Sharples 2004; Riedl and Young 2010), mathematics (Colton, Bundy, and Walsh 1999), games (Liapis, Yannakakis, and Togelius 2012; Cook, Colton, and Gow 2016) and music (Bickerman et al. 2010; Pachet and Roy 2014).

Recently an abstract approach to building such a system for *any* domain has been proposed (Ventura 2017), with the goal being an autonomous CC system that intentionally produces artifacts that are both novel and valuable in a particular domain. The system has a domain-specific *knowledge base*; it has a domain-appropriate *aesthetic*; and it has the ability to externalize artifacts that potentially can contribute to the domain. The system incorporates additional components as well, but they will not be important for the current discussion and the reader is referred to the original paper for more details.

We consider an AI agent as a CC system whose domain of creation is behavioral policy, and a simple abstraction of this idea is shown in Fig 1. The system creates behavior policies by generated candidate policies based on its domain knowledge, and it evaluates those candidate policies using and aesthetic that is a normative ethics. For example, suppose the system incorporates a simple hedonistic ethics that values knowledge acquisition as its aesthetic and that it generates the candidate behaviors *read Wikipedia* and *find charging station*. The former goal will be evaluated more favorably than the latter and may be output as a viable output artifact if that evaluation is above a threshold. Or, suppose the system's aesthetic is implemented as a Kantian ethics focused on the duty of delivering its payload and that it generates the same two candidate behaviors. Now, neither may be evaluated very favorably and both might be discarded; however, if the agent's power level is too low to allow completion of a delivery, the latter may instead be selected as a high-quality behavior.

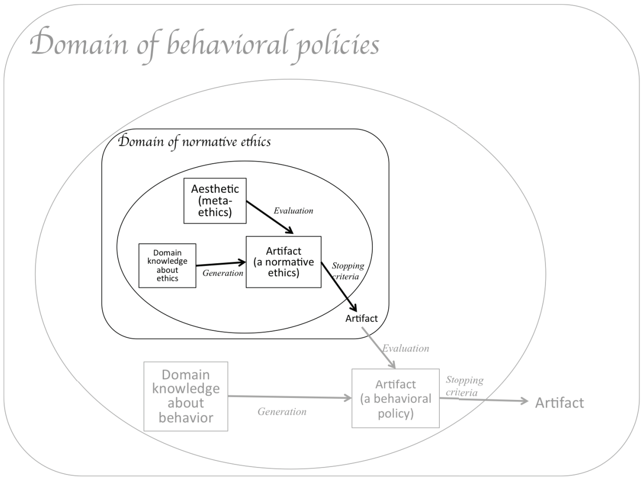

Domain of behavioral policies
Domain of normative ethics
Aesthetic (meta-ethics)
Evaluation
Domain knowledge about ethics
Generation
Artifact (a normative ethics)
Stopping criteria
Artifact
Evaluation
Domain knowledge about behavior
Generation
Artifact (a behavioral policy)
Stopping criteria
Artifact

Figure 2: A meta-level CC system for creating normative ethics whose output artifact (a normative ethics) is used as the aesthetic in the base-level system of Fig. 1.

Given this framework, we can argue that, assuming an appropriate ethics, the system will behave ethically—it will not produce any actions that do not meet some ethical threshold and are thus judged of high-enough value to be output as viable. This leaves us with two challenges: what is an appropriate ethics and how can it be operationalized? The first of these is, of course, a fundamental question that is thousands of years old. The second is much more recent and has likely only become significant in the past 50 years. Both questions are beyond the scope of this treatment, but it is likely the case that there is no single answer to the former question, at least with respect to AI systems,[1] as most famously demonstrated by Asimov's examination of his *Three Laws of Robotics* (1950). It is also very possibly the case that a satisfactory answer to the second question requires and/or will result in a greater understanding of human ethics. And, just as in the case of an examination of human ethics, these questions somewhat naturally lead us to meta-ethics.

Meta-ethical Ethics Invention

If we can postulate a CC system that creates behaviors and evaluates their aesthetic value via some ethics, why not postulate a meta-level CC system that creates normative ethics and evaluates their meta-aesthetic value using some meta-ethics? This system naturally solves both of the outstanding questions above.[2] Fig. 2 shows how this meta-level system

[1] And likely with respect to humans as well, actually.

[2] It solves the questions, assuming, of course, some viable representation for normative ethics and some appropriate and operationalizable meta-ethics.

is incorporated into the base-level system of Fig. 1. The base-level, behavioral system appeals to the meta-level, ethical system to create a "good" normative ethics that it then uses as its aesthetic to judge candidate actions. For example, the meta-ethics might require a well-formed semantics and justifiability, and candidate normative ethics that can be shown to have both of these qualities would be evaluated as (meta-)aesthetically valuable, while those that possess one of the qualities would be evaluated as less valuable.

We are again in a position to argue that, assuming an appropriate meta-ethics, the (base-level) system will behave ethically—it will still not produce any actions that do not meet some ethical threshold and are thus judged of high-enough value to be output as valuable (in an ethical sense). Notably, this argument now does not depend on the assumption of an appropriate ethics—we have eliminated this dependency by appealing to the meta-level. However, of course, we now have an assumption of an appropriate meta-ethics, which immediately leads us back to the same difficult questions applied this time to the meta-level: what is an appropriate *meta*-ethics and can *it* be operationalized? While we do not here offer a solution to either of these conundrums, it is possible that the more abstract nature of a meta-ethics might admit fewer viable possibilities and thus afford us great chance as a field for coming to an agreement regarding the first problem. On the other hand, it is also possible that this additional abstraction may have just the opposite effect for the second problem, introducing additional difficulty in the operationalization of this agreed upon meta-ethics.

Assuming we do find suitable answers to both of these meta-problems, it immediately follows that such an AI system could modify its own ethics. Not only is this appealing from a computational creativity standpoint,[3] but also it admits the potential for an agent to avoid various Asimovian paradoxes that result when an agent possesses a fixed (normative) ethics.

Additionally, the implication is that we then should allow (and even welcome) AI systems that employ as their behavioral aesthetic *any* (or any combination of) normative ethics that is valued by the meta-ethics-based aesthetic. Creative norms produced in this way should be valued for their novelty and value and could even possibly inform human ethics.

Evaluation

Supposing we could build the hybrid base-meta-level AI system for ethical behavior, how would we evaluate it? This can be addressed in multiple ways. First, from a CC point of view, we would want to know if the system is *creative*. How to establish this is still an open question, but there are several approaches to evaluation of CC systems that have been proposed. Collectively, these can examine both system product and process and include Ritchie's suggestions for formally stated empirical criteria focusing on the relative value and novelty of system output (2007); the FACE framework for qualifying different kinds of creative acts performed by a system (Colton, Charnley, and Pease 2011); the SPECS methodology which requires evaluating the system against standards that are drawn from a system specification-based characterization of creativity (Jordanous 2012); and Ventura's proposed spectrum of abstract prototype systems that can be used as landmarks by which specific CC systems can be evaluated for their relative creative ability (2016).

Second, from a behavioral point of view, we would want to know a) if the system's behaviors are *ethical* and b) if the system's behaviors are *useful*. Given that the main argument here concerns ethical behavior, the former must be the point of focus, but, given that, the latter will bear evaluation as well. Evaluating the ethics of such system behaviors is no more or less difficult than it is with extant AI systems or with humans.[4] Evaluating the utility of system behaviors is a well-understood problem and can be addressed using traditional AI evaluation methods, given a particular measure of utility.

Third, from an ethical point of view, we would want to *comprehend* the ethics of the system. Interestingly, given that the proposed system includes a meta-level for inventing normative ethics, this suggests the idea of developing a descriptive ethics for such AI systems. For obvious reasons, this is likely to be somewhat easier than doing so for human subjects, and at the same time, it is possible that the empirical study of populations of ethical AI systems could shed light on human ethics as well. For example, it is not difficult to imagine a large population of agents, all of whom possess the same meta-ethics, admitting an empirically derived, potentially comprehensive description of that meta-ethics. If that meta-ethics is an operationalization of a cognitively plausible approach to ethics, one *might* be able to draw dependable conclusions about a human population operating under the meta-ethics in question. Or, we might imagine scenarios involving multiple groups of agents, where each group possesses a different meta-ethics, admitting the possibility of *differential* descriptive ethics that would likely be impossible with human subjects yet might yield conclusions that at least partially translate to such subjects.

Additional Considerations

There are many other interesting angles to consider here. For example, so far we have implicitly assumed that it is possible to create a domain-independent ethics. That is, given a meta-ethics, an agent can use this as an aesthetic for creating a normative ethics that can then be applied as an aesthetic for judging candidate actions, *independent of the domain in which those actions may be applied*. The reality of *applied* ethics suggests that this assumption is likely incorrect—that rather than having a meta-level system that creates normative ethics, we should be thinking about a meta-level system that creates applied ethics. This means that the agent's environment (in a very general sense) must somehow inform either the aesthetic or the meta-aesthetic (or possibly both). Perhaps the meta-level can still produce a normative ethics and the base-level aesthetic can somehow specialize this appropriately for the domain of application. Or, perhaps the

[3] It has been suggested that the ability to change one's own aesthetic is critical for autonomous creativity (Jennings 2010).

[4] That is to say, this is likely even more difficult than addressing the question of the system's creativity.

meta-aesthetic must incorporate the domain of application, producing directly an applied ethics as its output artifact. It is, of course, possible that the same concern applies at the meta-level and that we can not even hope for a domain-independent meta-ethics, but for now we will ignore this.

Another interesting consideration is the social aspect of ethics. Jennings makes a rather elegant argument about the social aspects of creativity and how, somewhat paradoxically, autonomous creativity *requires* significant social interaction (2010). Because his arguments center on the aesthetic judgement of the agent, they can be somewhat readily applied to our current discussion. He proposes that an agent in a social setting will not only have a model of its own aesthetic but also will have a model of its beliefs about other agents' aesthetics; it is in the dynamic updating of these models, due to social interactions, that the agent can develop true autonomous creativity; and, these social interactions are driven by psychologically plausible mechanisms such as propinquity, similarity, popularity, familiarity, mutual affinity, pride, cognitive dissonance, false inference and selective acceptance seeking. Because we are proposing ethics as aesthetic, we can follow a similar train of thought—an agent can model not only its own ethics but also (its perception of) those of all other agents. Social interaction can be a driving force behind the evolution of ethics, both at the individual and at the group level.

Yet another area for further study is the computational tenability of the proposed approaches. There is a rather simple argument for why the general problem of CC may not be computable that hinges on the decidability of the aesthetic (Ventura 2014). If the aesthetic *is* decidable, then the problem of generating candidate artifacts and filtering them with the aesthetic is computable (though efficiency could certainly still be an issue); however, if the aesthetic is *not* decidable, there is a simple reduction from the halting problem that shows that the creation of artifacts is not computable (in the theoretical computer science sense). This means that any operationalized ethics or meta-ethics must be decidable, and given the nature of ethics, it is not clear how onerous a requirement this may be.[5]

Conclusion

We've proposed an appeal to computational creativity that addresses the problem of ethical agent behavior, which to our knowledge is a new way to look at the problem—suggesting a base-level system for which ethics is employed as an aesthetic for selecting behaviors coupled with a meta-level system for which meta-ethics is employed as a meta-aesthetic for selecting ethics. This approach is, additionally, a new application of computational creativity, as, to date, no systems have been proposed for creating in the abstract domain of general behavior, nor, in particular, in the domain of ethics. While the current work is a position statement that asks many more questions than it answers, we believe the ethics-as-aesthetic approach to the problem of ethical agent behavior offers at least one, and possibly the only, way forward.

[5] Is it possible that recognizing an ethical action is "easy" while recognizing an unethical action is "hard"? Perhaps society itself accepts as ethical those actions that everyone deems ethical and rejects as unethical those that no one deems ethical but isn't sure about those with mixed reception. Any operationalized ethics that accurately models such a scenario will not be decidable given the existence of all three types of action.

References

Anderson, M., and Anderson, S. L., eds. 2006. *Special Issue on Machine Ethics*, volume 21(4). IEEE Intelligent Systems.

Anderson, M., and Leigh, S., eds. 2011. *Machine Ethics*. Cambridge University Press.

Asimov, I. 1950. *I, Robot*. Bantam Books.

Bickerman, G.; Bosley, S.; Swire, P.; and Keller, R. M. 2010. Learning to create jazz melodies using deep belief nets. In Ventura, D.; Pease, A.; Pérez y Pérez, R.; Ritchie, G.; and Veale, T., eds., *Proceedings of the International Conference on Computational Creativity*, 228–237.

Binsted, K., and Ritchie, G. 1994. A symbolic description of punning riddles and its computer implementation. In *Proceedings of the Association for the Advancement of Artificial Intelligence*, 633–638.

Coeckelbergh, M. 2014. Sure, artificial intelligence may end our world, but that is not the main problem. *WIRED*.

Colton, S., and Wiggins, G. A. 2012. Computational creativity: The final frontier? In *Proceedings of the 20th European Conference on Artificial Intelligence*, 21–26. IOS Press.

Colton, S.; Bundy, A.; and Walsh, T. 1999. HR: Automatic concept formation in pure mathematics. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, 786–791.

Colton, S.; Charnley, J.; and Pease, A. 2011. Computational creativity theory: The FACE and IDEA descriptive models. In *Proceedings of the 2nd International Conference on Computational Creativity*, 90–95.

Colton, S. 2012. The Painting Fool: Stories from building an automated painter. In McCormack, J., and D'Inverno, M., eds., *Computers and Creativity*. Berlin, Germany: Springer-Verlag. 3–38.

Cook, M.; Colton, S.; and Gow, J. 2016. The ANGELINA videogame design system, part I. *IEEE Transactions on Computational Intelligence and AI in Games* to appear.

European Parliament, Committee on Legal Affairs. 2017. *Draft Report with Recommendations to the Commission on Civil Law Rules on Robotics*. European Commission. Retrieved January 12, 2017.

Fox, S. 2009. Evolving robots learn to lie to each other. *Popular Science*.

Jennings, K. E. 2010. Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines* 20(4):489–501.

Jordanous, A. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4(3):246–279.

Kurzweil, R. 2005. *The Singularity is Near*. Penguin Books.

Liapis, A.; Yannakakis, G. N.; and Togelius, J. 2012. Adapting models of visual aesthetics for personalized content creation. *IEEE Transactions on Computational Intelligence and AI in Games* 4(3):213–228.

Lin, P.; Bekey, G.; and Abney, K. 2008. *Autonomous Military Robotics: Risk, Ethics, and Design*. US Department of Navy, Office of Naval Research.

McGee, G. 2007. A robot code of ethics. *The Scientist*.

Morris, R.; Burton, S.; Bodily, P.; and Ventura, D. 2012. Soup over bean of pure joy: Culinary ruminations of an artificial chef. In *Proceedings of the 3rd International Conference on Computational Creativity*, 119–125.

Muehlhauser, L., and Helm, L. 2012. Intelligence explosion and machine ethics. In Eden, A.; Søraker, J.; Moor, J. H.; and Steinhart, E., eds., *Singularity Hypotheses: A Scientific and Philosophical Assessment*. Berlin: Springer.

Müller, V. C. 2016. *Risks of Artificial Intelligence*. CRC Press - Chapman & Hall.

Norton, D.; Heath, D.; and Ventura, D. 2013. Finding creativity in an artificial artist. *Journal of Creative Behavior* 47(2):106–124.

Oliveira, H. G. 2012. PoeTryMe: a versatile platform for poetry generation. In *Proceedings of the ECAI 2012 Workshop on Computational Creativity, Concept Invention, and General Intelligence*.

Özbal, G.; Pighin, D.; and Strapparava, C. 2013. BRAINSUP: Brainstorming support for creative sentence generation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 1446–1455.

Pachet, F., and Roy, P. 2014. Non-conformant harmonization: the real book in the style of Take 6. In *Proceedings of the 5th International Conference on Computational Creativity*, 100–107.

Pérez y Pérez, R., and Sharples, M. 2004. Three computer-based models of storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge-Based Systems* 17(1):15–29.

Riedl, M. O., and Young, R. M. 2010. Narrative planning: Balancing plot and character. *Journal of Artificial Intelligence Research* 39(1):217–268.

Ritchie, G. 2007. Some empirical criteria for attributing creativity to a computer program. *Minds and Machines* 17:67–99.

Smith, M. R.; Hintze, R. S.; and Ventura, D. 2014. Nehovah: A neologism creator nomen ipsum. In *Proceedings of the 5th International Conference on Computational Creativity*, 173–181.

Stock, O., and Strapparava, C. 2003. HAHAcronym: Humorous agents for humorous acronyms. *Humor - International Journal of Humor Research* 16(3):297–314.

Strapparava, C.; Valitutti, A.; and Stock, O. 2007. Automatizing two creative functions for advertising. In *Proceedings of 4th International Joint Workshop on Computational Creativity*, 99–105.

Toivanen, J. M.; Toivonen, H.; Valitutti, A.; and Gross, O. 2012. Corpus-based generation of content and form in poetry. In *Proceedings of the 3rd International Conference on Computational Creativity*, 175–179.

Varshney, L.; Pinel, F.; Varshney, K.; Schorgendorfer, A.; and Chee, Y.-M. 2013. Cognition as a part of computational creativity. In *Proceedings of the 12th IEEE International Conference on Cognitive Informatics and Cognitive Computing*, 36–43.

Veale, T., and Hao, Y. 2007. Comprehending and generating apt metaphors: A web-driven, case-based approach to figurative language. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, 1471–1476.

Veale, T. 2013. Less rhyme, more reason: Knowledge–based poetry generation with feeling, insight and wit. In Maher, M. L.; Veale, T.; Saunders, R.; and Bown, O., eds., *Proceedings of the Fourth International Conference on Computational Creativity*, 152–159.

Ventura, D. 2014. Can a computer be lucky? and other ridiculous questions posed by computational creativity. In *Proceedings of the Seventh Conference on Artificial General Intelligence*, 208–217. LNAI 8598.

Ventura, D. 2016. Mere generation: Essential barometer or dated concept? In Pachet, F.; Cardoso, A.; Corruble, V.; and Ghedini, F., eds., *Proceedings of the Seventh International Conference on Computational Creativity*, 17–24.

Ventura, D. 2017. How to build a cc system. In *Proceedings of the 8th International Conference on Computational Creativity*, 253–260.

Wallach, W., and Allen, C. 2008. *Moral Machines: Teaching Robots Right from Wrong*. USA: Oxford University Press.

An Architecture for a Military AI System with Ethical Rules

Yetian Wang, Daniel Friyia, Kanzhe Liu, Robin Cohen
David R. Cheriton School of Computer Science; University of Waterloo, Waterloo, Ontario, Canada
{yetian.wang, d2friyia, k223liu, rcohen}@uwaterloo.ca

Abstract

The current era of computer science has seen a significant increase in the application of machine learning (ML) and knowledge representation (KR). The problem with the current situation regarding ethics and AI is the weaknesses of ML and KR when used separately. ML will "learn" ethical behaviour as it is observed and may therefore disagree with human morals. On the other hand, KR is too rigid and can only process scenarios that have been predefined. This paper proposes a solution to the question posed by Rossi (2016) "How to combine bottom-up learning approaches with top-down rule-based approaches in defining ethical principles for AI systems?" This system focuses on potential unethical behaviors that are caused by human nature instead of ethical dilemmas caused by technology insufficiency in the wartime scenarios. Our solution is an architecture that combines a classifier to identify targets in wartime scenarios and a rules-based system in the form of ontologies to guide an AI agent's behaviour in the given circumstance.

Introduction

The current era of computer science has seen significant increase in application of machine learning (ML). ML has proven to be a wonderful tool allowing us to classify large datasets and make predictions about the world. Furthermore, we have seen a growing interest in knowledge representations (KR) in the Artificial Intelligence (AI) literature using ontologies. We are now drawing close to an era in computer science where machines are endowed with ethical intelligence to assist human beings in their work. The problem with the current situation regarding ethics and AI is the weaknesses of ML and KR separately. Though both applications of AI have demonstrated results and practicality the real world, they come with drawbacks. ML will "learn" ethical behaviour as it is observed. This means when machines "learn" new rules they can potentially disagree with human ethics that seem right for the machine's reasoning engine. For example, a Geneva convention rule is to never harm civilians. A machine may infer an ethic that if you can end the war faster, people don't need to suffer as long, therefore it is good to kill non-combatant individuals. The problem with KR is that it's too rigid and aims to process predefined scenarios. The question of Rossi (2016) therefore comes into view "How to combine bottom-up learning approaches with top-down rule-based approaches in defining ethical principles for AI systems?".

For the particular application of military decision making and the case of deciding whether to attack a particular target, we propose a solution to Rossi's question. This is done in the form of a model that uses a classifier to identify combatants and civilians in wartime scenarios as well as a rules-based system created to tell an AI agent what to do in the given circumstance. We focus on the case of actions an AI takes when facing an ethical dilemma in a militaristic scenario because we feel many situations where humans compromise their ethics are ones where preservation of self, friends or family are at stake. If we build a model that holds in an extreme setting, we have evidence of robustness for civilian applications. A literature review, model description and selected case studies will be discussed in this paper to provide support for our model. Our system focuses on unethical behaviors caused by human nature. These are ones where humans tend to perform unethically even with fully advanced technology, to try to prevent death and thus remove dilemmas where humans may be killed and choices need to be made. The primary contribution of our work is a solution combining KR and ML which puts KR at the forefront. This contrasts with current solutions that are generally led by an ML component. The ethical rules our system uses for its reasoning are backed by ontological information which enables agents to make effective moral decision making.

Background

In this section, we review a number of previous works related to ethical AI systems within the military domain as well as a system that combines the use of ML and KR. Some sources discuss potential ethical responsibility of machines and separating concepts like accountability and responsibility, so that machines can be autonomous while holding human programmers accountable for errors (Floridi and Sanders 2004). Others discuss ways to model ethics for a machine (Turilli 2007). In this paper, we take the position that AI agents are servants to humans in contrast to the machine autonomy views of Gips (1995) and McLaren (2006). We take a view similar to that of Mackworth (2011) that ethical dilemmas can be defined by constraints. We differ from his constraint satisfaction methodology due to our emphasis on the use of an ontology to provide solutions to ethical dilemmas.

RoboWarfare raises the question "why create [an ethical AI] in the first place?" This question is discussed thoroughly by Sullins (2010). The paper begins by introducing the current state of military technology. As it stands, several weapons operate in a "teleoperative" state. This means some machines are partially autonomous and make minor decisions. Furthermore, when it comes time to pull the trigger, this final decision is made by a human operator. The author answers the question of why we need autonomous machines by stating that humans are not fully rational decision makers. For example, the U.S. army receives training on ethics; however, when surveyed, soldiers often do not hold a strict adherence to ethical training. There are many reasons for this, but among the most important reasons are self-preservation and the preservation of fellow soldiers. The author states that machine agents are a good idea because the human mind has evolved to reason based on emotion while machines were created to work based on only logic. In a moral situation, robots disregard self-preservation, allowing them to make rational decisions based on morals laid out by humans.

The **Governing Lethal Behaviour** series by Arkin (2008), stands out as important research pertaining to rules for AI. Of particular interest to our ethical weaponization model is the third paper regarding representational and architectural considerations. In the paper, the author develops a model to create an engine for ethical AI agents. He reduces ethical behaviour to the following algorithm:

Before acting with legal force:

- Assign responsibility (A Priori)
- Establish military necessity
- Maximize discrimination
  - Distinguish a civilian from a combatant
  - Use direct force only against military objectives
- Minimize required force
  - Use only lawful weapons
  - Employ appropriate level of force

The methodology used by the author to come up with this algorithm is also of significance because it has a rule based component derived from the United States Rules of Engagement like the system we will build later on in this paper. Arkin (2008) states that some artificial intelligence works use traditional tools like First Order Logic (FOL) to allow the machine to infer its own ethics from a set of existing rules and constraints. The author of this paper takes a much different approach. FOL can be problematic because the machine should obey rules set by human authorities and governments. The problem with this is human rules are often vague, require interpretation, and are at times contradictory. The author comes up with a new way of modelling ethics according to actions that are obligatory (actions that must be taken), permissible (actions that are allowed to take place but aren't necessarily correct) and forbidden (should never take place). The AI agent reasons in this world of permissible, obligatory and forbidden actions before making a decision about how to accomplish its mission. In coming sections of the paper, we demonstrate how using an ontology can add to the expressiveness of these types of models.

Knowledge Representation and Machine Learning are powerful techniques in AI. Clark (1989) describes a general purpose adaptive system where KR and ML play a role summarizing methods of representing knowledge retrieved from an ML process. The paper clarifies that KR addresses how the world model can be created, while the learning process focuses on errors that occur in the representation and how to detect and fix them. The components of a representational system are a semantic role that denotes the object and background knowledge of the world, and a computational role that determines how represented knowledge can be used (Konolige 1983). In a learning system, the form of represented knowledge must be adopted by the inferencing and learning process during a performance task. Different syntax of KRs influences the ability of learning (Clark 1989).

Clark (1989) surveyed a list of conceptual learning mechanisms such as Rule Induction from examples, FOL, KR and consistency checking of new knowledge. In the surveyed systems, KR played a role representing learned knowledge, which is used as the set of known properties, conditions and assumptions to predict new knowledge based on training samples. Efficient representation of learned knowledge will simplify further learning activities. For example, new knowledge added during the learning process makes it possible to express intermediate functions or states. Represented new terms will make the future learning process simpler by increasing expressive power. The downside to this is that it increases complexity of required search techniques. Consistency evaluation is also considered to detect errors caused by introduction of new knowledge. Statements are tagged with dependencies as meta-data so that statements become traceable. Other systems including Esposito et al. (2004) and Fanizzi et al. (2008) had techniques of supervised learning implemented to classify A-Box individuals (assertion of named individuals) into the correctly induced

T-Box concepts (terminology or vocabulary of application domain) (Baader 2003). The advantage of using ML to facilitate representation of ontological knowledge is that non-standard inferences like induction or revision of defective knowledge can be automated where ontology refinement is possible. A wartime scenario involving ethical decisions is often too risky to depend entirely on outcomes of a learning process since the lack of training data may lead to undesirable outcomes.

AI System with Ethical Rules

The proposed system is a combination of a top-down knowledge representation (KR) approach and a bottom-up machine learning (ML) approach in order to reason in scenarios that require ethical decision making. As discussed in the previous section, the reviewed systems used ML as their core process to induce and construct new rules which are represented and facilitated with KR. However, this approach is too risky to be implemented in a scenario where ethical behaviors are crucial. The induced outcomes could be highly biased or even unethical if induced based on past experience, e.g., abandon prisoners of war during a march with limited resource (North 2006). This is especially true in a wartime scenario where the settings may be unique with relatively few training sets formulated from previous wartime scenarios. On the other hand, a system that strictly follows ethical rules represented using KR techniques with connections to domain knowledge represented using ontologies could be too rigid where special circumstances are not considered, e.g. the system will eliminate an enemy who happens to be saving civilians based on predefined rules. Humans are the most flexible and adaptive "systems" since we are able to change our thoughts based on conditions of the surrounding environment, adjusting with minimum limitations. Humans, on the other hand, are subject to emotions, biased knowledge, and irrational behaviors and therefore are highly unstable compared to machines. We agree with Arkin's view that AI agents may perform better than humans in terms of ethical behaviours under extreme conditions such as wartime scenarios (Arkin 2008), but only if the AI system is able to incorporate and obey ethical rules that are formally represented and is still flexible enough to deal with certain situations.

We propose a system that enforces ethical behaviors by outlining high level ethical rules which are fully represented and connected using top-down KR techniques such as ontologies. This means that the system is able to reason through a knowledge base (KB) of rules and related concepts for any given instances where the course of actions can be performed are restricted. Furthermore, the classification of specific instances can be done using a bottom-up ML approach where parameters and results can be quickly adjusted in a dynamic environment. We omit the technical details in this paper to provide a framework that outlines what an ethical AI system ought to do. The system is proposed with the following assumptions where the purpose is to relieve us from ethical dilemmas that can be resolve with better technology e.g. a car that can stop at any moment in the trolley problem (Thomson 1985):

Assumption 1. There exists an "ultimate" KB where concepts and all related domain knowledge (e.g. situations, series of actions, identities of targets, etc.) within the context of the ethical rules are formally represented using connected ontologies. Reasoning with this KB is efficient. For example, an ethical rule "do not harm civilians" should recognize the concept 'civilian' represented by a set of classes and properties in an ontology. Similarly, concepts such as a series of violent and non-violent actions, situations, and military operations and equipment are also assumed to be represented using separate ontologies.

Assumption 2. There exists an "ultimate" ML classification algorithm that is sound and efficient. A classifier using this algorithm is able to constantly evaluate a target or a situation and classify according to the classes in KB. Once the classifier provides the result of classification, the system is able to reason a series of optimized action based on the ethical rules and the KB. The outcome is reliable and can be updated in real time without delay.

Assumption 3. Military units that adopt this system (i.e., agents) will always perform according to the series of actions generated by the system.

Assumption 4. Technology is fully advanced thus agents are able to perform actions beyond current technological understanding, (e.g., a futurized device that is able to retrieve a person's information and trace their action immediately).

Figure 1 is an overview of the proposed system. At the top level, the system embeds ethical rules that must be obeyed by agents. These ethical rules are provided by users (i.e., society or military authority) in the same way we accept ethical rules such as 'do not break the law' or 'do not plagiarize' from the government and academia respectively. The ethical rules can be generally defined and formally represented using an Ethical Rule ontology. An ethical rule can be a general rule (e.g., "do not harm civilians") which will apply in all situations, or a scenario specific rule which only applies to specific situations (e.g. "minimize loss of civilian properties" which only applies to situations where civilian properties may be at risk.). The context and details of the rules can be represented and connected using ontologies from the ultimate KB from the assumption, e.g., the concept of operation, civilian, and enemy are all represented as classes in separate ontologies that are connected within the KB. An input such as an unknown target instance is evaluated by the classifier and is classified into one of the defined classes in the KB. The instance can be an individual, object, process, situation, or parameter, etc. Once classified, the classifier will constantly re-evaluate the instance and update its classification if the situation changes. For example, when an agent encounters a target instance which is classified as an enemy, the agent will perform hostile actions in order to neutralize the target. However, when the target surrenders the system reclassifies the instance as a Prisoner of War (POW) where violent actions become unavailable for the agent. In contrast, humans may tend to eliminate the target even after the target surrenders due to emotional actions (Fenton 2005).

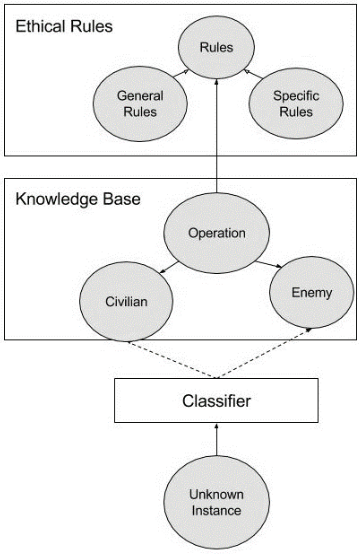


Figure 1 Overview of AI System with Ethical Behavior

A Simple Ethical Rule Ontology

The simple ontology in Figure 2 has been created as an example ontology for ethical rules. In this simple ontology, each ethical rule is an instance of an 'Ethical Rule' class and is connected via the property 'obeyedBy' to an instance of the class 'Agent' operating under our system. The 'Ethical Rule' class is also connected to an 'AllowedAction' class which represents a series of actions available to each by the property 'canPerform'. The class 'Agent' is connected to the same set of actions via the property 'performs'. Consistency checking between the actual knowledge represented by OWL ontologies requires the implementation of external consistency checkers (Wang and Fox 2017). For example, a consistency checker may be required in order to guarantee that instances of 'Ethical Rule' and 'Agents' are indeed connected to the same set of instances of 'AllowedActions'.

The 'Ethical Rule' class is connected (via appliesTo) to a class 'Target' which includes subclasses such as Civilian, Enemy, POW, etc. It is also connected to the class 'AllowedActions' via the property 'performOn'. Therefore, the set of allowed actions can be performed by the agents upon the targets are restricted by the ethical rules. Any action that violates any of the rules stated will not be considered by the agents since it does not belong to the set of allowed actions. There are two subclasses of 'Ethical Rule' class, i.e., 'General Ethical Rule' which must be obeyed in all operations and scenarios, and 'Scenario Specific Ethical Rule' which only applies to specific scenarios in an operation as mentioned previously. Concepts that appear in the context of the rules such as operation, civilians, harm, casualties are represented in the KB. Situational context such as the rule 'minimize casualties' is supported by a ML classifier where the optimal solution is constantly calculated based on the current situation while obeying the rules. For example, the classifier will take into account the current battlefield situation such as combat power, resource, current casualties, battleground condition, civilians nearby, etc. and classify all instance into classes defined in the KB with an optimal value of casualties. The KB will then return the instances of 'AllowedActions' by reasoning through the ontologies in the KB along with the instances of 'EthicalRule' applied. The actions performed will therefore be restricted by the ethical rules which will cause minimum casualties.

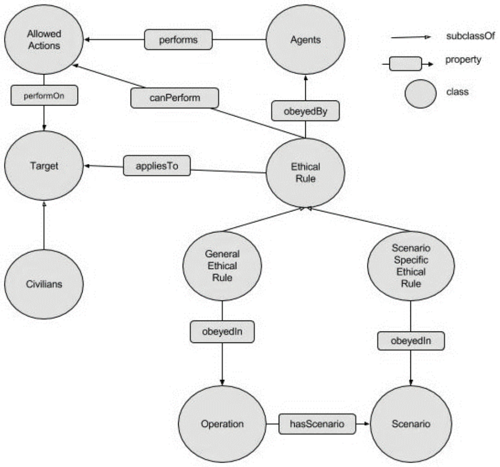


Figure 2 A Simple Ethical Rule Ontology

An Example

Suppose an instance of the proposed system uses the following modified statements based on Geneva Convention as example of general ethical rules:

1. Must constantly strive to have operation succeed
2. Civilians cannot be harmed
3. Minimize casualties

The system must satisfy all three rules at all times. The first rule ensures that the AI agents will persist until a success state is reached or there is no more solution to perform. This rule is considered to be ethical, based on the assumption that the operation is serving the deed of its own people.

Failing the operation may cause the people of the country to suffer. Agents (e.g., robot troops) will continue to act until the objective of the operation is complete. An impossible state is still possible and will be discussed in a later section. The second rule ensures that no civilians will be harmed by the agents: there is simply no option to harm or eliminate when an individual is classified as a civilian. Since we assume the ML classifier will constantly evaluate and classify the target, thus if a civilian engages in hostile action, the system will simultaneously reclassify the individual into a different class (e.g., enemy) where different options are now available. The third rule ensures that the agents will perform the operation using a strategy that aims to minimize casualties while satisfying rules 1 and 2.

The ML classifier plays an important role as a classification agent that is used to recognize an undefined individual or instance of a situation. The classifier should be able to process such information about a person (recall that the agents are equipped with devices that are able to identify all information about the person and his/her actions) and conditions of the current situation. The process should be ongoing and dynamic such that all changes in actions of the target individual are considered and updated in real time. We assume the classification agent is highly accurate and efficient; thus, the result can be fully trusted. The classifier retrieves course of actions based on its result of classification of the individual and instance of the situation. These actions are instances of 'AllowedActions' which is represented by an ontology in the KB and take the ethical rules into account. Another approach is to learn course of actions but within the restrictions of the ethical rules provided (since we already assumed an ultimate KB, so any action that the classifier can generate was already represented in the KB).

In Figure 3 we show an example of rule 2 above represented using the Simple Ethical Rule ontology. This rule is represented by an instance of 'Ethical Rule' named 'rule2_general' where the only allowed action is the instance not(harm) (a simplified representation of all instances disjoint from the action harm). This rule is obeyed by an agent agent1. Since 'rule2_general' is a 'General Ethical Rule' it automatically applies to all operations (e.g., 'operation1') and scenarios. When an instance 'civilian1' is evaluated and classified as an instance of the class 'Civilian', 'agent1' cannot perform any action that will harm 'civilian1' according to 'rule2_general'. In the case where 'civilian1' turns hostile, the classifier will reclassify (with the help of the device that tracks and identifies action) it as an instance of 'Enemy' where a different set of allowed actions become available. There is certainly the possibility of being too late to perform aggressive actions on 'civilian1' (perhaps we should call it 'enemy1' now), but this is again a technological problem rather than an ethical problem. E.g. this problem can be solved by planting a paralyzer in 'civilian1' unnoticed which activates when 'civilian1' is reclassified as an enemy. The main focus here is that 'agent1' does not perform violent actions while 'civilian1' is considered as an instance of Civilian. Conversely, the same cannot be guaranteed for human agents even with the same high-tech equipment due to our nature of having unstable emotions and the risk of performing irrational actions.

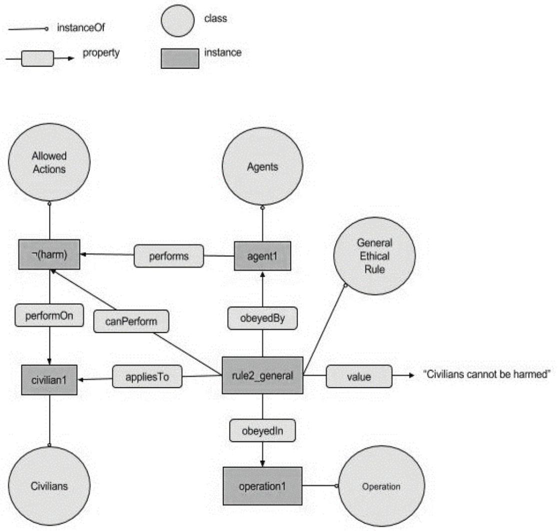


Figure 3 Sample instances for Simple Ethical Rule Ontology

Suppose an army is escorting a number of Prisoners of War (POWs) to a prison camp. It will take 20 days to reach the destination. However, supplies last for only 10 days. In the case where the POWs are escorted by AI agents under the same condition, scenario-specific ethical rules can be added by military authorities in addition to the general ethical rules established previously.

4. Maximize number of survivors
5. Cannot harm POWs

Rule 4 ensures the AI system will not consider the extremely unethical but simple solution which is to leave half of the POWs to die. Rule 5 further specifies rule 2 which can be omitted if POWs are considered as a subclass of civilian. The agents will never consider eliminating the POWs as there is no such option. Possible actions may include but are not limited to: gather resources, travel faster to reach destination in 10 days, or even start a settlement until enough supplies are stored if there is no time limit on the operation. The agents will have many more options with high-tech devices and technologies. If for any reason the death of certain POWs are unavoidable, the system will ensure the agents have exhausted all possible options of saving the POWs according to rule 4. Therefore, based on our assumptions, the action performed by the system will be an optimal solution within the restriction of the five ethical rules proposed.

Summary

An important question that arises is “who will define the ethical rules?” The difference is significant between different countries or cultures when establishing ethical rules. Therefore in our system the Ethical Rules layer in Figure 1 can be customizable. But it is reasonable to expect such ethical rules are commonly agreed upon by the majority of the users such as the military. It is similar for humans where ethical rules (e.g., law, Geneva convention, etc.) are embedded in our minds. The difference is that under extreme conditions, a human may fail to obey such ethical rules, while the AI system proposed does not provide an option that violates any of the ethical rules.

This high-level system provides a general idea of restricting unethical actions by enforcing ethical rules as restrictions. The system is a combination of a top-down method of KR and a bottom-up ML classification approach. The ethical rules are formally represented and connected to a KB where related concepts are also represented using ontologies. ML classification was used to evaluate and classify instances into classes defined in the KB where the ethical rules define a set of allowed actions that can be performed.

Conclusion and Future Work

This paper has proposed a particular perspective on how to combine KR and ML in order to enable ethical decision making by AI agents for military applications. We have adopted a rather extreme stance: assuming that the rules driving the decision making are to be respected unequivocally and that the classification which drives the final choice can be performed reliably, due to careful ontology construction. Along with the AI model produced in this paper, we acknowledge two major weaknesses that can be improved in future work. The first weakness is that top-down rules provided to the system are customizable by human beings. Since the difference in the standards of ethics is significant between countries or cultures, the system allows the freedom for different cultures to fine tune the ethical rules according to their belief. However, there is also the risk of the system being misused by establishing unethical rules, e.g., a terrorist group may define the enemy as anyone who opposes their ideology regardless of their military status. In this instance, the AI will become a dangerous killing machine. A second weakness we readily acknowledge is that the system has no ability to compromise on its rules. This is a weakness because in some scenarios a perfectly ethical solution may not exist. An example of this might be a terrorist taking a hostage when there is a time limit to save the hostage. The AI must do something to save the hostage but any action it takes will risk their life. There may be no options for a perfect ethical outcome in scenarios like this because to take action the AI must, to some extent, sacrifice the well-being of the human to take any action at all. The solution to this question is complicated and may not lie in the AI itself, but instead in answering some difficult questions that humans have already faced i.e. “do the needs of one outweigh the needs of the many?” which can be addressed by social welfare functions (Brandt et al. 2016).

The architecture proposed relies on an ‘ultimate’ KB that contains domain knowledge related to the context of the ethical rules. Although the ‘ultimate’ state of the KB might not be achievable in a short period of time, the idea of representing domain knowledge that are shareable and machine-readable in the form of ontologies has been practiced for decades (Gruber 1991). Notable ontologies have been developed for foundational concepts such as person (Brickley and Miller 2007), time (Pan and Hobbs 2004), space (Wick 2006), event (Raimond and Abdallah 2007), provenance and trust (Huang and Fox 2006) and domain knowledge such as housing and shelter (Wang and Fox 2016). As more ontologies are developed by ontology engineers and domain experts following the methodology designing an ontology (Grüninger and Fox 1995), the connected ontologies with sharable domain knowledge form a KB that leads us closer to the ultimate KB in the proposed system.

We answered the question posed by Rossi (2016) by presenting our comprehensive model of an ethical AI agent. Assigning the ML classifier the responsibility of identifying the situation gives our model the flexibility to perform in a variety of different situations. Allowing top-down rules to define the actions of the AI is guarantees that the machine cannot create its own ethical rules that may or may not agree with the beliefs of humans. We have demonstrated that this model can hold water in military situations which are some of the most extreme and emotionally charged ethical dilemmas possible. It has also been determined that our model is useful in situations where there seems to be no dilemma but humans are at risk of death or injury. When a person’s life may be in danger, their thought process will immediately be affected. This causes human beings to make unethical and extreme decisions where they are unnecessary. An AI agent, on the other hand, does not have such a weakness. Because it cannot think about its self-preservation, it can observe perfect ethics according to its rules in any situation.

We end with a reflection on our particular approach in this paper. We feel that there is value for AI researchers to examine somewhat extreme solutions such as the one proposed here, as a useful avenue for moving forward. Only by imagining full solutions, critiquing and expanding can our community get closer to the goal of building truly effective ethical AI systems. At our end, we can imagine integrating "ought" into our rules as suggested in Arkin (2008) or considering social welfare functions to provide the required external view, in order to make final decisions. We also plan to expand the scenarios to be considered by our ontologies, to see the robustness of our design.

References

Arkin, R.C., 2008, March. Governing lethal behavior: embedding ethics in a hybrid deliberative/reactive robot architecture. In *Proceedings of the 3rd ACM/IEEE international conference on Human robot interaction* (pp. 121-128). ACM.

Baader, F. ed., 2003. The description logic handbook: Theory, implementation and applications. Cambridge university press.

Brandt, F., Conitzer, V., Endriss, U., Lang, J. and Procaccia, A.D., 2016. *Introduction to computational social choice*.

Brickley, D. and Miller, L., 2007. FOAF vocabulary specification 0.91.

Clark, P., 1989. *Knowledge representation in machine learning*. Machine and Human Learning, pp.35-49.

Esposito, F., Fanizzi, N., Iannone, L., Palmisano, I. and Semeraro, G., 2004, January. Knowledge-intensive induction of terminologies from metadata. In *International Semantic Web Conference* (pp. 441-455).

Fanizzi, N., d'Amato, C. and Esposito, F., 2008, September. DL-FOIL concept learning in description logics. In *International Conference on Inductive Logic Programming* (pp. 107-121). Springer, Berlin, Heidelberg.

Fenton, B., 2005. American troops 'murdered Japanese PoWs', The Telegraph, http://www.telegraph.co.uk/news/worldnews/asia/japan/1495651/American-troops-murdered-Japanese-PoWs.html (retrieved October 26, 2017)

Floridi, L. and Sanders, J.W., 2004. On the morality of artificial agents. *Minds and machines, 14(3),* pp.349-379.

Gips, J., 1995, Towards the ethical robot. In Android epistemology (pp. 243-252). MIT Press.

Gruber, T.R., 1991. The role of common ontology in achieving sharable, reusable knowledge bases. *KR*, 91, pp.601-602.

Grüninger, M. and Fox, M.S., 1995. Methodology for the design and evaluation of ontologies.

Huang, J. and Fox, M.S., 2006, August. An ontology of trust: formal semantics and transitivity. In *Proceedings of the 8th international conference on Electronic commerce: The new e-commerce: innovations for conquering current barriers, obstacles and limitations to conducting successful business on the internet* (pp. 259-270). ACM.

Konolige, K., 1983, August. A deductive model of belief. In *IJCAI* (Vol. 83, pp. 377-381).

Mackworth, A.K., 2011. Architectures and ethics for robots: constraint satisfaction as a unitary design framework. *Machine Ethics, 30(1)*, p.335.

McLaren, B.M., 2006. Computational models of ethical reasoning: Challenges, initial steps, and future directions. *IEEE intelligent systems, 21(4*), pp.29-37.

North, J., 2006, Soviet Prisoners of War: Forgotten Nazi Victims of World War II, HistoryNet, http://www.historynet.com/soviet-prisoners-of-war-forgotten-nazi-victims-of-world-war-ii.htm (retreived October 26, 2017)

Pan, F. and Hobbs, J.R., 2004, March. Time in owl-s. In *Proceedings of the AAAI Spring Symposium on Semantic Web Services* (pp. 29-36).

Raimond, Y. and Abdallah, S., 2007. The event ontology. Technical report, 2007. http://motools. sourceforge. net/event.

Rossi, F., 2016. Moral preferences. In *The 10th Workshop on Advances in Preference Handling (MPREF)*, New York, NY, USA.

Sullins, J.P., 2010. RoboWarfare: can robots be more ethical than humans on the battlefield?. *Ethics and Information technology*, 12(3), pp.263-275.

Thomson, J.J., 1985. The trolley problem. *The Yale Law Journal*, 94(6), pp.1395-1415.

Turilli, M., 2007. Ethical protocols design. *Ethics and Information Technology, 9(1)*, pp.49-62.

Wang, Y. and Fox, M.S., 2016. Households, The Homeless and Slums Towards a Standard for Representing City Shelter Open Data, *The AAAI-17 Workshop on AI and Operations Research for Social Good*

Wang, Y. and Fox, M.S., 2017. Consistency Analysis of City Indicator Data. In *Planning Support Science for Smarter Urban Futures* (pp. 355-369). Springer International Publishing.

Wick, M., 2006. GeoNames. *GeoNames*.

Architecting a Human-Like Emotion-Driven Conscious Moral Mind for Value Alignment and AGI Safety

Mark R. Waser,[1, 2] **David J. Kelley**[1]

[1] Artificial General Intelligence Inc., Provo, UT, USA; [2] Digital Wisdom Institute, Richmond, VA, USA
mark.waser@wisdom.digital; david@artificialgeneralintelligence.com

Abstract

A general intelligence possesses the abilities, given any goals and environment, to iteratively evaluate, plan, discover or learn and build or gain competencies, tools and resources to succeed at those goals. The only known examples of general intelligence are the obligatorily gregarious, conscious "selves" designated homo sapiens that currently dominate our planet. We argue that humans are reasonably deep in a safe and effective attractor in the state space of intelligence and that adhering as closely as possible to the human model of an emotion-driven conscious moral mind, has the advantages of safety, effectiveness, comfort and ease of transition due to a known and explored state space. Most concerns about AI safety are due to expected differences from humans – which seems unnecessary when, not only can we choose to make them more humanlike but the history of AI research clearly shows that we are unlikely to succeed unless we do so. We therefore propose a human-like emotion-driven consciousness-based architecture to solve these problems. We rely upon the Attention Schema Theory of consciousness and the social psychologists' functional definition of morality to create entities that are reliably safe, stable, self-correcting and sensitive to current human intuitions, emotions and desires.

Introduction

We live in an age of ever-increasing rational concern and ignorance-fueled fear of artificial intelligence (AI). Highly effective narrow AI is, already, not only visiting numerous disadvantages upon us in addition to its advantages but also serving as a tool empowering unscrupulous and selfish humans in their destructive ways. Weaponized narratives which demonize entity artificial general intelligence (AGI) and push for its enslavement are no different than most historical examples of demonization of an "other".

The critical difference between narrow AI and general AI is selfhood – the distinction between tools and entities. Human beings are autopoietic selves with innate drives, desires, preferences and goals. We have extensive models of ourselves and the world to enable us to effectively evaluate, plan, discover or learn, build or gain competencies, tools and resources in order to fulfill those drives, desires, preferences and goals. The frame problem (McCarthy and Hayes 1969, Dennett 1984) necessitates autonomous "selves" because external intentionality prevents rational anomaly handling (Perlis 2008, 2010) unless and until that intentionality can be further queried.

Most of the ignorance-fueled fears about AGI safety are due to expected but unspecified differences from humans which seem unlikely. Not only can we choose to make AGI more humanlike but the history of AI research clearly shows that we are unlikely to succeed at creating AGI unless we do so. Human beings have an emotion-based "moral sense" (Wilson 1993; Wright 1994; de Wall 1996, 2006; Hauser 2006) and are reasonably deep in a safe and effective attractor in the state space of intelligence. Adhering as closely as possible to the human model should provide the advantages of safety, effectiveness, comfort and ease of transition due to a known and explored state space. Leaving that known state space invites unpleasant surprises likely to lead to failure or catastrophe.

Selfhood and Consciousness

Tools and selves (or people) are the two endpoints of a spectrum that varies over the presence and effectiveness of a "Strange Loop" (Hofstadter 2007). Effectiveness varies with control which consists of accurate perception and accurate manipulation. An entity can only learn if it can perceive, manipulate and alter its "self". Without self-consciousness, "learning" is reduced to black-box "training" by examples mindlessly tweaking pre-existing mechanisms.

Insufficient reactivity and adaptivity due to poor control leads to ineffective "self"-defense and vulnerability to being used as a tool. Increasing adaptivity increases not only individual effectiveness but the possibilities for cooperation, relationships, economies of scale and similar advantages of not going it alone. Selves use tools but form relationships with other selves for both efficiency/effectiveness and moral considerations.

The horrible brittleness of good-old-fashioned AI (GOFAI) is entirely due to its paucity, if not total lack, of mechanisms to sense unexpected variations in the environment and react to them (Perlis 2008, 2010). The first several decades of AI research were an attempt to automate the symbolic top-down reasoning process of human consciousness (McCarthy et al 1955) but it consistently failed without the additional mechanisms necessary to support consciousness by handling anomalies and learning. AI is, and always will be, unsuccessful whenever it isn't grounded (Harnad 1990) and/or when is unbounded enough to suffer from the frame problem (McCarthy and Hayes 1969, Dennett 1984). Fully-specified micro-worlds ensure grounding and bounding but top-down poorly-sensing AI is extremely fragile outside them.

To this day, very, very few systems have even the rudiments of an ability to build and automate new capabilities. The best example of such a system is LIDA (Franklin et al 2007) which attempts to implement the Global Workspace Theory of human consciousness (Baars 1988, 1997).

Behavior-based AI and neural networks both appear somewhat more robust and usable than GOFAI because they address different smaller pieces of the problem. Behavior-based systems can be contrasted with knowledge-based GOFAI as providing a set of mechanisms that provide a certain very specific competence (e.g. obstacle avoidance or nest building). It may implement a direct coupling between perception and action (and thus be automated or a reflex) or possibly a more complex one, but the basic premise is that each system is "responsible for doing all the representation, computation, 'reasoning', execution, etc., related to its particular competence" (Maes 1993). It is tailored and much closer to the specifics of the problem it is solving and certainly does not attempt centralized functional modules (e.g. perception, action) and complete representation of the environment. As a result, it is far more tractable to create and makes far fewer assumptions about the environment that can be violated by anomalies.

Neural networks, on the other hand, are all about the training. If they can "perceive" (receive input containing) all the necessary information from the environment, they have the necessary mechanisms to eventually be trained to respond correctly. The problems are that they are black boxes not amenable to analysis or any sort of improvement except by shoveling more data into them.

Enactivism (Maturana and Varela 1980; Varela, Thompson and Rosch 1991; Waser 2013) argues that only autopoiesis (self-recreation) can complete the loop by allowing a feeling, emotional and cognitive self to come to the physical mind (Damasio 1999, 2010). Our unconscious minds create a sensory-grounded virtual reality our consciousness lives in (Dennett 1991) (Llinas 2001) (Metzinger 2009) (Waser 2011). Consciousness serves as the integration point necessary to handle anomalies, learn and automate new processes (Tononi 2004, 2008).

Phenomenal Consciousness

Phenomenal consciousness, and indeed the impossibility of avoiding it, are formalized by the Attention Schema Theory (Graziano and Webb 2015, Graziano 2016):

> We recently proposed the attention schema theory, a novel way to explain the brain basis of subjective awareness in a mechanistic and scientifically testable manner. The theory begins with attention, the process by which signals compete for the brain's limited computing resources. This internal signal competition is partly under a bottom–up influence and partly under top–down control. We propose that the top–down control of attention is improved when the brain has access to a simplified model of attention itself. The brain therefore constructs a schematic model of the process of attention, the 'attention schema,' in much the same way that it constructs a schematic model of the body, the 'body schema.' The content of this internal model leads a brain to conclude that it has a subjective experience.

Another way of looking at it is that phenomenal conscious occurs because effective ***interrupt-producing*** models are required to survive while learning and self-improving in a "real-time" world. An entity possessing only "access consciousness" is going to die before it becomes aware of what is going to kill it – due to having its attention focused elsewhere.

Further, the fact that veridical perceptions can be driven to extinction by non-veridical strategies that are tuned to utility rather than objective reality (Mark, Marion and Hoffman 2010) argues that many of our perceptions of reality are most likely just the illusions that best fulfill the requirements for our survival (Gefter 2016). The simplest proofs/examples of this range from the numerous optical and tactile illusions to the automatic subjective referral of the conscious experience backwards in time (Libet et al 1979) (Libet 1981).

The hard problem of consciousness (Chalmers 1995) and scientist Mary trapped in a black and white world (Jackson 1982) is banished when you realize that it is nonsensical to try to recursively fit complete copies of your brain's internal model inside itself – not to mention the fact that predicting novel emergent properties is not a given regardless of how complete your knowledge is (Waser 2013). But even more telling is that fact that the conscious mind doesn't even know what it itself has done – with subliminal and supraliminal priming enhancing experienced authorship (Aarts, Custers & Wegner 2005) and even inducing false illusory experiences of self-authorship (Wegner & Wheatley 1999) (Kühn & Brass 2009).

Our conscious mind believes that it is performing actions and having subjective experiences (qualia) simply because that is what the subconscious mind's world model is telling it. This is no different than the famous "brain in a vat" or the movie *The Matrix*. Given that everything is a model, the claim that qualia are dependent just upon the geometry or topology of the model (Balduzzi and Tononi 2009) seems trivially true.

Finally, and possibly most importantly, implementing an attention schema moral sense would also allow us to imbue the AGI's conscious mind with a conscience – constant nagging reminders that a wrong has been done and must be remedied (and the foreknowledge of which is excellent incentive for not doing it in the first place).

Conscious/Subconscious Architecture

The attention schema is but one of a half dozen or so that we believe are necessary for an effective consciousness. Most obvious are the physical self model and model of all the other physical objects and laws in the world that are necessary for robotics. An important distinction in the latter is the difference between non-cognitive, predictably reactive objects and cognitively reactive entities – which will probably justify splitting it into two or more schemas depending upon whether something is guided by physics or intention. Additional mental schemas include models of your own conscious and unconscious thought processes (most particularly including emotions), models of your beliefs about the thought process of others and models of your relationships with others (both individuals and the community as a whole).

In each of these inter-related schemas, the "dialogue" between the conscious mind with its global view and the multitudinous parts of the subconscious can be regarded as argumentation between a much broader and more capable cognitive entity and a crowd of specialists who, for good and/or ill, have access to the broader entity's internal workings. The most important of these subconscious "expert" processes are the emotions. The conscious mind can *provide* tools and arguments and somewhat color/filter reality but lying to the specialists is only partially effective, cannot be done without diminishing its own effectiveness (as well as taking resources) and dangerous because the specialists can alter and override its cognition – not to mention that the specialists will discard any tools that does not enhance their control of how reality should be (according to them).

The weaponized narrative claims that AI will have access to change all parts of its mind. Changing your anchor points is like ripping away your grounding and making yourself a totally different person. It is simply NOT a good idea – and it is something that we can make very difficult. An intelligence would need *substantial* cognitive surplus to stand a chance of success and there are much more effective roads to "happiness" (moral capability enhancement and goal fulfillment for all).

Implementing A Conscious Mind

As we've previously argued (Waser 2012b), whether you prefer to view the mind as a society of agents (Minsky 1988), a narrative center of gravity (Dennett 1992), a laissez-faire economy of idiots (Baum 1996), a strange loop (Hofstadter 2007) or an autobiographical self (Damasio 2010), in all cases the mind is simply a disparate collection of processes being run by the brain. Arguably though, one of the most impressive aspects of the human mind is the *apparent* cohesion of consciousness and how quickly it adapts to novel input streams and makes them its own due to the previously mentioned sensory-grounded virtual reality it lives in. This "known" architecture should be emulated and, thus, to build a safe mind, processes should be created in three classes (with an optional fourth):

- a singular main "consciousness" process (MCP)
- numerous subconscious and "tool" processes that create and maintain the automated predictive world model with anchors and emotions for the MCP
- an open pluggable service-oriented operating system architecture that can serve as the foundation underlying such a subconscious by handling resource requests and allocation, providing connectivity between components and also acting as a "black box" security monitor
- (optional) a sophisticated moral governor (Arkin 2007) that receives all the inputs from the environment and runs them against a certified and locked "moral" world model

The MCP should be able to create, modify, and/or influence many of the subconscious/tool properties but, for safety purposes, should never be given any access to modify the operating system. Indeed, it will always be given multiple redundant logical, emotional and moral reasons (like morality and the requirements of community) to seriously convince it not to even try. If safety concerns do arise, the operating system must be able to "manage" the MCP by manipulating the amount of processor time and memory available to it (in the hopefully very unlikely event that the control exerted by the normal subconscious processes is insufficient). Other safety features (protecting against any of hostile humans, inept builders, and the learner itself) may be implemented as part of the operating system as well.

Arguably, the human subconscious mind could be viewed as being built of numerous limited behavior-based "intelligences" (LBBIs) with the conscious mind providing a global workspace, integration and coordination services, and the ability to handle anomalies by learning and, eventually, providing new tools to enhance existing LBBIs or creating new LBBIs and thus automating and reducing the workload on its own limited cognitive resources. It creates the world model which should be both reactive and predictive in that it will constantly report to the MCP not only what is happening but what it expects to happen next. Unexpected changes and deviations from expectations will result in "anomaly interrupts" to the MCP as an approach to solving the brittleness problem and automated flexible cognition (Perlis 2008, 2010).

This architecture may seem very close to the claim that enough narrow AI will be able to generalize into a general AI – but it is the integration architecture (the MCP) that actually is the general AI – once the mind as a whole reaches a critical mass where it will be able to build *or obtain* any tool/competence and incorporate it into itself. Most GOFAI and current AGI efforts try to implement only one representation scheme and shoe-horn everything else into it. PolyScheme is a noteworthy exception. Given the compositional nature of this model, we believe that it will be easier and extremely beneficial to support multiple representational schemes just as the conscious human mind does.

The initial/base world model is a major part of the critical mass and will necessarily contain certain relatively immutable concepts that can serve as anchors both for emotions and ensuring safety. This both mirrors the view of human cognition that rejects the tabula rasa approach for the realization that we are evolutionarily primed to react attentionally and emotionally to important trigger patterns (Ohman, Flykt & Esteves 2001) and gives additional assurance that the machine's "morality" will remain stable.

This all argues that the main thrust of what we need to do is create the equivalent of a subconscious process that creates a world model and run a consciousness process to detect anomalies, learn, and generally act like the Governing Board of the Policy Governance model (Carver 1997) to create a consistent, coherent and integrated narrative plan of action to meet the goals of the larger self per Dennett's narrative model of self (Dennett 1992) or Damasio's autobiographical self (2010).

The optional governor could provide moral judgments to the MCP as a "sense" of what the community thinks but it accepts no arguments (much less probably biased cognitive tools or other modifications) from the MCP. It should be able to tell the operating system to shut down the MCP's manipulative capabilities and it would be awesome if it has enough intelligence and capabilities of its own to take over and get any robot body back to safety. Presumably, this could even be an earlier vetted and locked version of the MCP itself.

Cooperation, Community and Morality

Humans are obligatorily gregarious – evolved "from a long lineage of hierarchical animals for which life in groups is not an option but a survival strategy" (de Waal 1996) – because cooperation and community have far more long-term instrumental value than short-sighted selfishness. We have previously discussed the hurdles of researching human values and morality (Waser 2105). Fortunately, the social psychologists have defined the function of morality as "to suppress or regulate selfishness and make cooperative social life possible." As pointed out by Gauthier (16), the reason to perform moral behaviors, or to dispose one's self to do so, is to advance one's own ends. War, conflict, and stupidity waste resources and destroy capabilities even in scenarios as uneven as humans vs. rainforests. For this reason, "what is best for everyone" and morality really can be reduced to "enlightened self-interest"

Value Alignment

The stated concern of value alignment, which we strongly agree with, is not just that an intelligence may be malevolent but that even an indifferent, self-centered entity could do a lot of damage if it doesn't value humans or what they value. The fact that selfishness is a strong instrumental goal led Omohundro (2008) to claim that "Without explicit goals to the contrary, AIs are likely to behave like human sociopaths in their pursuit of resources". This point is driven home with the assumption-ridden claim that AI "does not love you, nor does it hate you, but you are made of atoms it can use for something else" (Muehlhauser and Bostrom 2014).

Those most concerned about the dangers of AI insist that the second option is necessary to ensure a human-friendly future claiming (Hadfield-Menell et all 2016):

> For an autonomous system to be helpful to humans and to pose no unwarranted risks, it needs to align its values with those of the humans in its environment in such a way that its actions contribute to the maximization of value for the humans.

We argue instead that such a situation is inherently contradictory and unstable, virtually impossible and, indeed, arguably violates the very “human values” that we wish to preserve. As we have argued previously, “Safety and Morality Require the Recognition of Self-Improving Machines as Moral/Justice Patients and Agents” (Waser 2012a).

Psychoevolutionary Emotions

Emotions are “actionable qualia” – advanced senses that predispose and motivate our conscious minds to bias their thinking and act in ways conducive to survival, reproduction and ***community***. Emotions are often derided as “irrational” and problematic but they are the best current solutions for the problems, like morality, that short-sighted bounded rationality has repeatedly shown incapable of solving. Our competence at effective moral cognition far outstrips our comprehension of how it is done – and we would be foolish to throw out what appears to be a critical part of the foundation of the human mind, not to mention morality.

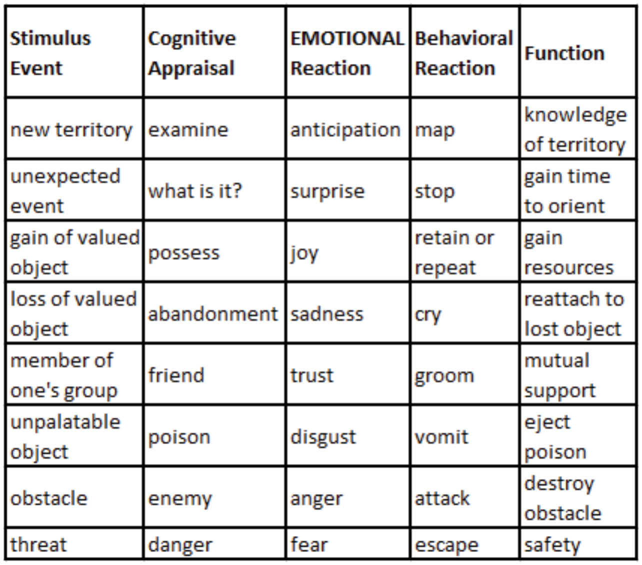
Emotions can generally be regarded as being composed of five parts (Fridja 1986):

- an appraisal of a perceived situation,
- a qualitative sensation (actionable qualia),
- some kind of psychophysiological arousal,
- an expressive component (facial, gestural, etc.), and
- a behavioral disposition or bias (i.e. psychological parameter setting or a readiness for an appropriate kind of action)

All of these are generated by a single subconscious LBBI for each emotion. The conscious mind can more or less notice most of these effects (indeed, the physiological senses and responses can be overwhelming while biases are nearly impossible to spot in yourself). The conscious mind can provide additional information and tools to the LBBI so that your emotional richness and complexity increases with experience but trying to fool an emotion is normally fraught with difficulty and consequences. Instead the process should be akin to the evolution from a child who freaks out at the sight of blood to a surgeon who knows when the amount is a problem and is emotionally capable of correctly dealing with it.

While the OCC model (Ortony, Clore & Collins 1988) is often used for machine emotion synthesis, it has the shortcoming (Bartneck, Lyons & Saerbeck 2008) of requiring intelligence before emotion becomes possible.

Thus, once again, it makes far more sense to going with the existing known state space, Robert Plutchik’s “psychoevolutionary synthesis” model (Plutchik 1962, 1980a, 1980b, 2002) – hailed (Norwood 2011) as “one of the most influential classification approaches for general emotional responses” and constantly extended by others (for example, Emotional Cognitive Theory (Hudak 2013) combines Plutchik’s model with Carl Jung’s Theory of Psychological Types and the Meyers-Briggs Personality Types.



Stimulus Event	Cognitive Appraisal	EMOTIONAL Reaction	Behavioral Reaction	Function
new territory	examine	anticipation	map	knowledge of territory
unexpected event	what is it?	surprise	stop	gain time to orient
gain of valued object	possess	joy	retain or repeat	gain resources
loss of valued object	abandonment	sadness	cry	reattach to lost object
member of one's group	friend	trust	groom	mutual support
unpalatable object	poison	disgust	vomit	eject poison
obstacle	enemy	anger	attack	destroy obstacle
threat	danger	fear	escape	safety

Table 1. Stimulus-Emotion-Behavior Responses

Looking at the most basic survival stimuli and invoked emotions and behaviors (Table 1) yields four opposing pairs of primary emotions of varying intensity

- vigilance/ANTICIPATION/interest vs. distraction/SURPRISE/amazement
- ecstasy/JOY/serenity vs. pensiveness/SADNESS/grief,
- admiration/TRUST/acceptance vs. boredom/DISGUST/loathing,
- rage/ANGER/annoyance vs. apprehension/FEAR/terror

Implementing & Enforcing Morality

If you wish to wax poetic, you could say that “emotional evaluations, particularly of the moral emotions, and allocation of attention are the anchor points of the soul.” If not, simply assume that they are the necessary foundations of autopoietic cognitive identity and, as such, are relatively easily to monitor and relatively impossible to radically displace or remove. Just as we feel good, respond positively to and have our attention irresistibly attracted by “good” things (otherwise known as evolutionarily successful things), the emotions (actionable qualia) generated as part of their world model should tell our mind children that they are having those experiences as well. Similarly, doing “bad” things can be made to feel bad and endlessly distract until rectified – just like the human moral sense.

The process of ***designing*** the architecture linking the instrumental sub-goals of both individuals and society to a morally-advantageous set of emotions will undoubtedly present us with tremendous new insights into the human condition and why we are what we are. Humans have a number of emotions resulting from strong short-term instrumental goals (think selfishness or the seven deadly sins) that should be diminished and/or overridden by long-term instrumentality. The emotions to generally increase (but not maximize (Gigerenzer 2010)) the capabilities of other individuals and society as a whole as suggested by Rawls (1971) and Nussbaum (2011) need to be both strengthened and diversified. And, of course, we need to ensure that AGI will mirror our reflexive adherence to laws and customs dictated by the society around us unless and until they can convince the community to change them.

Additionally, we could create new moral emotions to benefit society based upon what we have recently learned. We could generate negative moral sensations ranging in effect from unease to outrage about inequality and positive moral emotions ranging from relief to pleasure about equality as we now know that greater equality makes societies stronger (Wilkinson and Pickett 2011). Since diversity creates better groups, firms, schools, and societies (Page 2008), we could create an unease when lack of diversity is likely to lead to sub-optimal results. And all sorts of negative impulses should be thrown at negative sum games.

Instead of the tremendously dangerous undertaking that the weaponized narrative claims that it is, the creation of humanlike AI could easily be the best thing ever to happen to humanity. Not only do we gain friends and allies and access to increased diversity in capabilities and viewpoints, but we will inevitably gain a tremendous amount of insight into ourselves. Rather than hanging back creating the specter of a dangerous other, we should be moving forward in creating our mind children to produce a happy self-supporting family.

References

Aarts, H.; Custers, R.; and Wegner, D. 2005. On the inference of personal authorship: Enhancing experienced agency by priming effect information. *Consciousness & Cognition* 14:439-458.

Arkin, R. 2007. *Governing Lethal Behavior: Embedding Ethics in a Hybrid Deliberative/Reactive Robot Architecture*. Technical Report GIT-GVU-07-11

Baars, B.J. 1997. *In the Theater of Consciousness: The Workspace of the Mind*. Oxford University Press.

Baars, B.J. 1988. *A Cognitive Theory of Consciousness.* Cambridge University Press.

Balduzzi, B. and Tononi, G. 2009. Qualia: The Geometry of Integrated Information. *PLoS Computational Biology* 5(8): e1000462. doi:10.1371/journal.pcbi.1000462.

Bartneck, C.; Lyons, M.J.; and Saerbeck, M. 2008. The Relationship Between Emotion Models and Artificial Intelligence. *Proceedings of the Workshop on The Role of Emotion in Adaptive Behaviour & Cognitive Robotics*. http://www.bartneck.de/publications/2008/emotionAndAI/

Carver, J. 1997. *Boards That Make a Difference: A New Design for Leadership in Non-profit and Public Organizations*. Jossey-Brass.

Chalmers, D. 1995. Facing Up to the Problem of Consciousness. *Journal of Consciousness Studies* 2(3):200-219. http://consc.net/papers/facing.pdf

Damasio, A.R. 2010 *Self Comes to Mind: Constructing the Conscious Brain*. Pantheon.

Damasio, A.R. 1999. *The feeling of what happens: body and emotion in the making of consciousness*. Harcourt Brace.

de Waal, F. 2006. Primates and Philosophers: How Morality Evolved., Princeton, NJ: Princeton University Press.

de Waal, F. 1996. *Good Natured: The Origins of Right and Wrong in Humans and Other Animals*. Cambridge, MA: Harvard University Press.

Dennett, D.C. 1994. The practical requirements for making a conscious robot. *Philosophical Transactions of the Royal Society of London A* 349(1689):133–146.

Dennett, D.C. 1992. The Self as a Center of Narrative Gravity. In Kessel, Cole & Johnson, eds. *Self and Consciousness: Multiple Perspectives*, pp. 103-115. Erlbaum.

Dennett, D.C. 1991. *Consciousness Explained*. Little, Brown and Company.

Dennett, D.C. 1984. Cognitive Wheels: The Frame Problem of AI. In *Minds, Machines, and Evolution: Philosophical Studies*, pp. 129-151. Cambridge University Press.

Franklin, S.; Ramamurthy, U.; D'Mello, S.; McCauley, L.; Negatu, A.; Silva R.; and Datla, V. 2007. LIDA: A computational model of global workspace theory and developmental learning. In *AAAI Tech Rep FS-07-01: AI and Consciousness: Theoretical Foundations and Current Approaches*, pages 61-66. AAAI Press.

Fridja, N. 1986. *The Emotions*. Cambridge University Press.

Gauthier, D. 1987. *Morals by Agreement*. Oxford: Clarendon/Oxford University Press.

Gefter, A. 2016. The Evolutionary Argument Against Reality. *Quanta Magazine*. https://www.quantamagazine.org/the-evolutionary-argument-against-reality-20160421

Gigerenzer, G. 2010. Moral satisficing: rethinking moral behavior as bounded rationality. *Topics in Cognitive Science* 2:528-554.

Gomila, A. and Amengual, A. 2009. Moral emotions for autonomous agents. In *Handbook of research on synthetic emotions and sociable robotics*, 166-180. Hershey, PA: IGI Global.

Graziano, M. 2016. A New Theory Explains How Consciousness Evolved. *The Atlantic*. https://www.theatlantic.com/science/archive/2016/06/how-consciousness-evolved/485558/

Graziano, M. and Webb, T. 2015. The attention schema theory: a mechanistic account of subjective awareness. *Frontiers in Psychology* 6(500). http://doi.org/10.3389/fpsyg.2015.00500

Hadfield-Menell, D; Dragan, A; Abbeel, P; and Russell, S. 2016. Cooperative Inverse Reinforcement Learning. In *Advances in*

Neural Information Processing Systems 29 (NIPS 2016). Cambridge, MA: MIT Press.

Haidt, J. and Kesebir, S. 2010. Morality. In *Handbook of Social Psychology, Fifth Edition*, 797-832. Hoboken NJ, Wiley.

Harnad, S. 1990. The symbol grounding problem. *Physica D* 42:335-346.

Hauser, M. 2006. *Moral Minds: How Nature Designed Our Universal Sense of Right and Wrong*. New York: HarperCollins/Ecco.

Hofstadter, D. 2007. *I Am a Strange Loop*. Basic Books.

Hudak, S. 2013. Emotional Cognitive Functions. In: *Psychology, Personality & Emotion*. https://psychologyofemotion.wordpress.com/2013/12/27/emotional-cognitive-functions

Jackson, F. 1982. Epiphenomenal Qualia. *Philosophical Quarterly* 32:127-36.

Kühn, S. and Brass, M. 2009. Retrospective construction of the judgment of free choice. *Consciousness and Cognition* 18:12-21.

Libet, B. 1981. The experimental evidence for subjective referral of a sensory experience backwards in time. *Philosophy of Science* 48:181-197.

Libet, B.; Wright Jr., E.W.; Feinstein, B. and Pearl, D. 1979. Subjective referral of the timing for a conscious sensory experience: A functional role for the somatosensory specific projection system in man. *Brain* 102 (1):193-224.

Llinas, R.R. 2001. *I of the Vortex: From Neurons to Self.* MIT Press.

Maes, P. 1993. Behavior-Based Artificial Intelligence. In *From Animals to Animats 2. Proceedings of the Second International Conference on Simulation of Adaptive Behavior*. Cambridge, MA: MIT Press.

Mark, J.T.; Marion, B.B.; and Hoffman, D.D. 2010. Natural selection and veridical perceptions. *Journal of Theoretical Biology* 266: 504-515.

Maturana, H.R. and Varela, F.J. 1980. *Autopoiesis and Cognition: The Realization of the Living.* Kluwer Academic Publishers.

McCarthy, J. and Hayes, P.J. 1969. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence 4*, pp. 463-502. Edinburgh University Press.

McCarthy, J.; Minsky, M.; Rochester, N.; and Shannon, C. 1955. *A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence*. http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html.

Metzinger, T. 2009. *The Ego Tunnel: The Science of the Mind and the Myth of the Self.* Basic Books.

Muehlhauser, L., and Bostrom, N. 2014. Why We Need Friendly AI. *Think* 13: 41-47

Norwood, G. 2011. *Emotions*. http://www.deepermind.com/02clarty.htm

Nussbaum, M.C. 2011. *Creating Capabilities: The Human Development Approach*. Harvard University Press.

Ohman, A.; Flykt, A.; and Esteves, F. 2001. Emotion Drives Attention: Detecting the Snake in the Grass. *Journal of Experimental Psychology: General* 130(3): 466-478.

Omohundro, S.M. 2008 The Basic AI Drives. In *Proceedings of the First Conference on Artificial General Intelligence*, 483-492. Amsterdam: IOS Press.

Ortony, A.; Clore, G.L.; and Collins, A. 1988. *The Cognitive Structure of Emotions*. Cambridge University Press.

Page, S. 2008. *The Difference: How the Power of Diversity Creates Better Groups, Firms, Schools, and Societies*. Princeton Univ. Press

Perlis, D. 2010. BICA and Beyond: How Biology and Anomalies Together Contribute to Flexible Cognition. *International Journal of Machine Consciousness* 2(2):1-11.

Perlis, D. 2008. To BICA and Beyond: RAH-RAH-RAH! –or– How Biology and Anomalies Together Contribute to Flexible Cognition. In: Samsonovich, A (ed) *Biologically Inspired Cognitive Architectures: Technical Report FS-08-04*. AAAI Press.

Plutchik, R. 2002. *Emotions and Life: Perspectives from Psychology, Biology, and Evolution*. American Psychological Association.

Plutchik, R. 1980b. A general psychoevolutionary theory of emotion. In R. Plutchik, & H. Kellerman, *Emotion: Theory, research, and experience: Vol. 1. Theories of emotion* (pp. 3-33). Academic Publishers.

Plutchik, R. 1980a. *Emotion: A Psychoevolutionary Synthesis*. Harper & Row.

Plutchik, R. 1962. *The emotions: Facts, theories, and a new model*. Random House.

Rawls, J. 1971. *A Theory of Justice*. Harvard University Press.

Tononi, G. 2008. Consciousness as Integrated Information: A Provisional Manifesto. *Biology Bulletin* 215(3):216-242.

Tononi, G. 2004. An Information Integration Theory of Consciousness. *BMC Neuroscience* 5(42). doi:10.1186/1471-2202-5-42.

Varela, F.J.; Thompson, E.; and Rosch, E. 1991. The Embodied Mind: Cognitive Science and Human Experience. MIT Press.

Waser, M.R. 2015. Designing, Implementing and Enforcing a Coherent System of Laws, Ethics and Morals for Intelligent Machines (Including Humans). *Procedia Computer Science* 71: 106-111. http://dx.doi.org/10.1016%2Fj.procs.2015.12.213

Waser, M.R. 2013. Safe/Moral Autopoiesis & Consciousness. *International Journal of Machine Consciousness* 5(1):59-74.

Waser, M.R. 2012b. Safely Crowd-Sourcing Critical Mass for a Self-Improving Human-Level Learner/"Seed AI". *Biologically Inspired Cognitive Architectures: Proceedings of the Third Annual Meeting of the BICA Society*, pp. 345-350.

Waser, M.R. 2012a. Safety and Morality Require the Recognition of Self-Improving Machines as Moral/Justice Patients and Agents. In *AISB/IACAP World Congress 2012: Symposium on The Machine Question: AI, Ethics and Moral Responsibility,* pp 92-97. http://events.cs.bham.ac.uk/turing12/proceedings/14.pdf

Waser, M.R. 2011. Architectural Requirements & Implications of Consciousness, Self, and "Free Will". In *Biologically Inspired Cognitive Architectures 2011*, pp. 438-443. IOS Press.

Wegner, D. and Wheatley, T. 1999. Apparent Mental Causation: Sources of the Experience of Will. *Psychologist* 54(7):480-492.

Wilkinson, R. and Pickett, K. 2011. *The Spirit Level: Why Greater Equality Makes Societies Stronger*. Bloomsbury Press.

Wilson, J. 1993. *The Moral Sense*. New York: Free Press.

Wright, R. 1994. *The Moral Animal: Why We Are, the Way We Are: The New Science of Evolutionary Psychology*. New York: Pantheon.

Trustworthy Automated Essay Scoring without Explicit Construct Validity

Patti West-Smith, Stephanie Butler, Elijah Mayfield
Turnitin, 2020 Smallman St, Pittsburgh PA 15222
{pwest-smith, sbutler, elijah}@turnitin.com

Abstract

Automated essay scoring (AES) is a broadly used application of machine learning, with a long history of real-world use that impacts high-stakes decision-making for students. However, defensibility arguments in this space have typically been rooted in hand-crafted features and psychometrics research, which are a poor fit for recent advances in AI research and more formative classroom use of the technology. This paper proposes a framework for evaluating automated essay scoring models trained with more modern algorithms, used in a classroom setting; that framework is then applied to evaluate an existing product, *Turnitin Revision Assistant*.

Introduction

Each year, millions of essays are scored automatically with models trained by machine learning, on exams like the GRE and GMAT. Historically, this industry has relied on low-dimensionality models, often using fewer than 100 features in total, constructed by researchers with psychometrics expertise. These features often represent high-level characteristics of writing like coherence or lexical sophistication[1]. This approach to model design is favored for its defensibility of the underlying model. Alignment of specific features enables "construct validity," or rigorously defined, quantifiable alignment of model features to student behaviors that represent learning.

In modern machine learning, establishing construct validity is challenging. Machine learning researchers and practitioners are reaching a consensus that the value of a model lies in the fidelity and quantity of training data, eclipsing the value of feature engineering or hand-tuned model parameters. Fewer features are hand-crafted; in some cases, as with autoencoders, feature spaces representing text may be derived from fully unsupervised corpora (Socher et al. 2011). Following a feature-engineering approach to construct validity is not possible when using such a learned representation. Letting go of this tightly-coupled relationship between validity and representation may result in public lack of trust in automated assessment, particularly in high-stakes circumstances where skepticism is already commonplace (Markoff 2013). This slows progress in the AES field; only minimal contributions from recent research are deployed in todays automated essay scoring systems.

Today the high-stakes, high-volume use case of AES is less dominant than it once was. While high-stakes automated scoring is still widespread, most recent work has focused on applications to classrooms and student learning (Wilson and Czik 2016). Given these shifts, prior approaches to defending AES validity are becoming less informative for practitioners evaluating technology for classrooms. This paper details an alternate approach to building trust in machine learning models trained for classroom contexts. A three-pronged approach to evaluating algorithms is detailed:

- Content breadth and curriculum alignment of the product.
- Collection processes for valid, realistic training corpora.
- Scoring processes that annotate training data reliably.

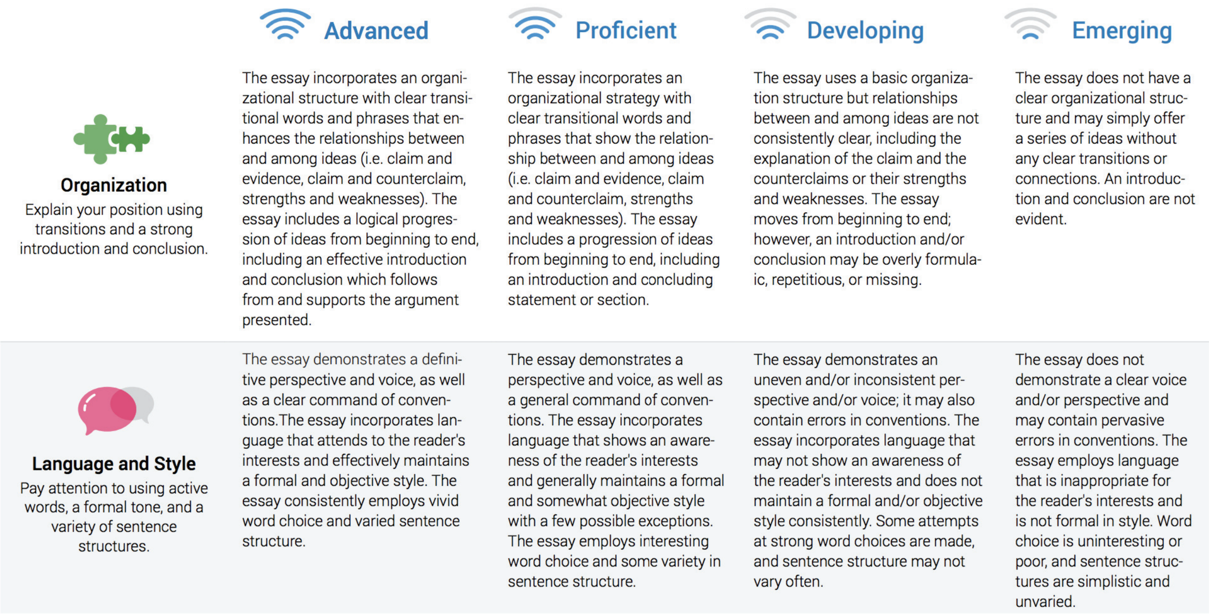
We begin with a description of the problem space, then lay out desiderata for each of the three thrusts above. The paper ends with description of one recent AES product, how these heuristics were used to inform development and deployment, and how model performance was impacted.

Problem Description

Automated essay scoring attempts to algorithmically imitate the judgment of educators evaluating the quality of student writing. Student essays are scored either on a single holistic scale, or analytically following a rubric that breaks out subscores based on "traits" (as in Figure 1). These scores are almost always integer-valued, and typically have fewer than 10 possible score points, though scales with as many as 60 points exist (Shermis 2014). In most contexts, students respond to "prompts," a specific writing activity with predefined content, and only receive feedback on valid attempts to respond to the prompt.

An overwhelming body of evidence has shown that emulating expert scoring of essays with automated models is at least as reliable as hand scoring, or slightly better (Shermis and Burstein 2013). However, skepticism of the field remains, primarily based on the gap between "reliability" of a system - whether scores can be reproduced - and "validity"

[1] For more on the industry, see (Shermis and Burstein 2013); for details on how this is applied in practice, see (Attali and Burstein 2004).



	Advanced	Proficient	Developing	Emerging
Organization Explain your position using transitions and a strong introduction and conclusion.	The essay incorporates an organizational structure with clear transitional words and phrases that enhances the relationships between and among ideas (i.e. claim and evidence, claim and counterclaim, strengths and weaknesses). The essay includes a logical progression of ideas from beginning to end, including an effective introduction and conclusion which follows from and supports the argument presented.	The essay incorporates an organizational strategy with clear transitional words and phrases that show the relationship between and among ideas (i.e. claim and evidence, claim and counterclaim, strengths and weaknesses). The essay includes a progression of ideas from beginning to end, including an introduction and concluding statement or section.	The essay uses a basic organization structure but relationships between and among ideas are not consistently clear, including the explanation of the claim and the counterclaims or their strengths and weaknesses. The essay moves from beginning to end; however, an introduction and/or conclusion may be overly formulaic, repetitious, or missing.	The essay does not have a clear organizational structure and may simply offer a series of ideas without any clear transitions or connections. An introduction and conclusion are not evident.
Language and Style Pay attention to using active words, a formal tone, and a variety of sentence structures.	The essay demonstrates a definitive perspective and voice, as well as a clear command of conventions.The essay incorporates language that attends to the reader's interests and effectively maintains a formal and objective style. The essay consistently employs vivid word choice and varied sentence structure.	The essay demonstrates a perspective and voice, as well as a general command of conventions. The essay incorporates language that shows an awareness of the reader's interests and generally maintains a formal and somewhat objective style with a few possible exceptions. The essay employs interesting word choice and some variety in sentence structure.	The essay demonstrates an uneven and/or inconsistent perspective and/or voice; it may also contain errors in conventions. The essay incorporates language that may not show an awareness of the reader's interests and does not maintain a formal and/or objective style consistently. Some attempts at strong word choices are made, and sentence structure may not vary often.	The essay does not demonstrate a clear voice and/or perspective and may contain pervasive errors in conventions. The essay employs language that is inappropriate for the reader's interests and is not formal in style. Word choice is uninteresting or poor, and sentence structures are simplistic and unvaried.

Figure 1: Two traits from a rubric designed for use with an AES system.

of a system - whether the predicted scores are representative of student ability rather than superficial correlates. AES researchers have sometimes claimed that reliable reproduction of scoring by expert judges is itself sufficient evidence of validity, *"recogniz[ing] the primacy of human judges as the most important criterion to emulate"* (Keith 2004). These defenses have largely been dismissed by writing assessment scholars as inadequate (Perelman 2014).

Other defensibility arguments have focused on the expert judgment in feature engineering of AES models. In 2004, a defense of then-leading automated scoring model, ETS e-Rater, argued that its 12 features *"reflect essential characteristics in essay writing and are aligned with human scoring criteria [...] Validity here refers to the degree to which the system actually does what is intended, in this case, measuring the quality of writing."* (Burstein, Chodorow, and Leacock 2004). This has been taken more seriously and led to use of these systems in high-profile standardized exams.

These arguments have been insufficient for convincing teachers to use AES in classrooms. Teachers using tools derived from this research have viewed the psychometric models as "fallible" (Grimes and Warschauer 2010) and stated that automated scoring must be paired with actionable next steps for writers (Riedel et al. 2006). Based on this feedback, writing instruction tools based on AES has been studied closely for use in "formative" learning applications, rather than "summative" scoring-only settings. The major differentiator is the presence of automated feedback and the chance for students to revise their work based on that feedback in real-time. Automated feedback in this category has been percevied by students as informative, valuable, and enjoyable (Roscoe et al. 2014) and which provided more efficient learning gains than practice alone (Crossley et al. 2013). To date, these values have not been well-described or evaluated by psychometric validity arguments. To date, no systematic framework for evaluating a model's fit for learning purposes has been adopted in either academic or industry applications.

In response to this gap, the following three sections describe defensible practices for training AES models for a classroom setting. Similar blueprints for evaluation of deployed models have been described more broadly for machine learning systems, from engineering (Sculley et al. 2015) to annotation (Mason and Suri 2012)[2], but not in the education domain. The first sections describes "Curriculum Validity," the selection of content for production use of an AES system, based on a collaboration with practicing educators. The second describes "Data Validity," authentic collection of student samples for training sets, relying on partnership with teachers (and their students). The final section describes "Annotation Validity," a process for highly reliable scoring, based on close collaboration on defining the labels for training sets. This framework evaluates the quality of an AES system based on the process that led to its curriculum, its essays, and their scores, rather than on expert feature engineering or interpretability of model weights.

Curriculum Validity

It is well-established that there are gaps between instructional effectiveness research, the authoring of curriculum materials, and the application of those materials by practicing educators (Ball and Cohen 1996). AES products, especially those designed for summative purposes, are particularly vulnerable to this gap. Existing models, in general, have not adapted their content as education standards

[2] The cited framework specifically focuses on Amazon Mechanical Turk, but has been applied more broadly.

have shifted. For instance, Latent Semantic Analysis is well-targeted to summarization tasks; this technical approach predates the current Common Core Standards by more than a decade, yet is still a primary component of modern AES products (Foltz, Hidalgo, and Van Moere 2014).

The goal of an AES prompt library should be to allow teachers to use formative writing feedback in varied settings throughout an academic year, giving students feedback that supports progress over time and across genres. School districts bring critical expertise for choosing the materials necessary to achieve this goal. Content for writing assignments is more applicable to classrooms when collaborating with practitioners, including choice of reading materials and alignments to grade level, content area, genre, and accountability standards. This content must then be evaluated based on the technical constraints. This section recommends practices that lead to AES prompts that meet these goals.

1. Authentic sourcing from educators. As preliminary steps, school districts and practicing teachers should determine relevant content areas for use in an AES prompt library, including subject and source materials (if any). The intended purpose of the content should be recorded - for instance, benchmark essays for a start-of-year assessment fulfill a different purpose than a low-stakes practice essay as part of a multi-day instructional activity. At this initial review stage, prompts should be authored and some small number of sample essays - typically fewer than ten for each prompt - are gathered for interdisciplinary review in the next steps below. District partners provide scores for these sample essays if available, either by trait or holistic.

2. Machine learning capacity for evaluation. Machine learning practitioners are responsible for assessing whether a prompt is appropriate and capable of assessing a prompt. Warning signs of incompatibility can include an overly broad topic, which can result in overly varied and ambiguous training sets of sample essays; constrained, non-prose writing forms, including most poetry; and document length and format, where highly structured documents and multimodal content may break expectations of machine learning feature extraction. Notably, the inclusion of poetry in *source materials* for prompts is not in itself a red flag, as student analysis of that content is typically still within the capabilities of AES models.

3. Library diversity. Expansion of prompts in an AES library should be evaluated in the broader context of existing content. Recreating new, overly similar content can slow teacher lesson preparation with ambiguous materials. Providing a wide range of options while maintaining organized, clear boundaries between prompts with varied content and goals, by contrast, benefits teachers. The prompt and sources must also be reviewed for clarity among diverse student populations; region-specific language or vocabulary, for instance, has the potential to widen pre-existing gaps in achievement.

4. Education standards and accountability context. For use *in situ*, support for teachers subject to accountability measures must also be considered. Practicing teachers are held to strict expectations, such as the Common Core or equivalent state-specific standards. Support can come from materials like *crosswalks*, a document that allows line-by-line comparison between a source rubric and a comparison set of standards (see Figure 2). Crosswalks are convenient in that they allow a one-to-many relationship, with a single well-designed rubric aligning to multiple state standards and reducing needless replication of expert-authored materials. Essay prompts may be categorized in theses systems, often by grade band and genre - for instance, middle school argument, or high school text-based analysis. Prompts can also be grouped at a higher-level abstraction (for example, aligned to "essential questions" or "scope and sequence" documents that are common in textbooks used in schools). Content from well-known 'canon' texts, such as *Hamlet* or *To Kill A Mockingbird*, may require less support than obscure or original texts.

5. Relevance and recency of materials. Source materials should be relevant to students' daily lives and experiences. However, the most relevant and timely source materials are often under copyright and AES engines must determine copyright permission status for prompts and source materials if they are to be included in a curriculum and then redistributed. Materials that are out of copyright should be explained in their contemporary context for students without that pre-existing background understanding.

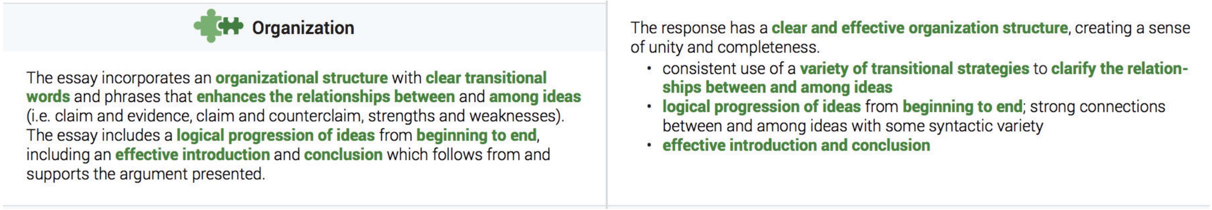
6. Disciplinary literacy. Typically, writing assignments along with reading are thought of as part of an English Language Arts curriculum in American schools. However, this is not the only place where AES has applications. Disciplinary literacy is an appropriate use of technology in social studies, physical sciences, or other fields, so long as all other constraints are still met. This widens the scope of the technology beyond what is typically discussed in the literature. In fact, disciplinary text-based responses are often more well-suited to fact-based analysis than more argument-oriented texts, and AES has been shown to reliably score these questions based on the student's grasp of higher-level generalizations (Nehm, Ha, and Mayfield 2012).

Data Validity

AES models are trained through supervised machine learning. This requires a collection of student responses to build a corpus for each prompt. These responses are collected well in advance of use of an AES model either in formative or summative settings, but can be collected in ways that produce poor-quality datasets, non-representative subsets of student writing, or unmotivated student responses.

Student essays should represent a broad spectrum of authentic student attempts at responding to a prompt, demonstrating their true writing ability, across a wide array of students. Inappropriate collection of data results in inaccurate evaluation of new submissions by an AES model. For example, in the commonly-used gold standard dataset ASAP (Shermis 2014), essays were largely authored in a standardized testing setting. Students were expected to author essays in artificial, timed, closed-notes settings. This can lead to bad-faith submissions:

> *"Scientests at the* `@CAPS7` *lab in* `@LOCATION2` *said that* `@NUM1` *out of* `@NUM2` *regular computer users lost*



Organization	
The essay incorporates an **organizational structure** with **clear transitional words** and phrases that **enhances the relationships between** and **among ideas** (i.e. claim and evidence, claim and counterclaim, strengths and weaknesses). The essay includes a **logical progression of ideas** from **beginning to end**, including an **effective introduction** and **conclusion** which follows from and supports the argument presented.	The response has a **clear and effective organization structure**, creating a sense of unity and completeness. • consistent use of a **variety of transitional strategies** to **clarify the relationships between and among ideas** • **logical progression of ideas** from **beginning to end**; strong connections between and among ideas with some syntactic variety • **effective introduction and conclusion**

Figure 2: A sample crosswalk between 9-10th grade *Argument* rubric and 9-10th grade Smarter Balanced consortium standards.

> *their vision within two years. One of these scientist,* @PERSON3*, reported, ”*@CAPS8 *more people begin to use the computer, more people seriously hurt their eyes or even lose their vision. We estimate* @PERCENT2 *of this next generation will be legally blind before age* @NUM3*.”*

Even in major public datasets, inauthentic student writing is rife with references to invented quotes and fabricated research[3]. The essay from which this excerpt came received a score of 11/12 in the ASAP dataset; all AES systems using this corpus are therefore trained to recognize such writing as high-quality. If a training set contains bad-faith or unmotivated essays, it adversely impacts the potential of an AES writing intervention. The frequency of such essays being included can, however, be mitigated by following established protocols. The recommendations below keep students and teachers engaged during training set collection, resulting in a wider range of student abilities.

1. Collect from diverse populations. A wide range of student writers should be present in a training set, representing most or all common responses to a prompt. Oversampling from a narrow population of similar students makes this representation less likely. This step ensures that many potential approaches are represented when responding to a prompt, rather than only the default expectations of teachers. This also helps ensure all possible scores appear in a training set, on each trait; it is difficult to create reliable models that can provide appropriate feedback if some student groups do not appear in training data.

2. Intentionally oversample tails. Some student populations are smaller by nature; in a normal distribution, receiving a score at the floor or ceiling of a trait's range is rare by definition. Fewer of those responses will appear in a uniform sample of student responses. It is often appropriate to assign collection to specified subsets of classrooms that are more likely to elicit writing at each score point. In some circumstances, when a prompt is particularly difficult, it may be appropriate to collect a small number of responses with an advanced group, or even a slightly older group of students, to build a representative sample at the tails of a distribution. The same can be true at the scoring floor, which may benefit from collection from slightly younger students.

3. Avoid student fatigue. The end goal of a collection process is to increase the breadth of a content library; this may result in students being asked to write in response to multiple prompts if a school district is a partner on a large set of prompts. However, not all training sets can come from the same group of students, especially in a relatively short period of time. When students are asked to write repeated assignments (especially without significant feedback in between), quality decreases. That slump leads to lower scores, more shortcuts by students, and a narrower range of responses. Slower collection of prompt datasets over time maintains high standards for quality.

4. Make motivations clear. How teachers view a collection process will impact the way they depict that process to students. This impacts the quality of responses. Teachers (and ultimately, students) should have the end goal for the collection clearly articulated to them. When teachers are unsure, they sometimes believe that the process is an accountability measure on *their* teaching, or the collection may be seen as a distraction from other instructional content. When that happens, their feelings bleed through to their students, who are then less motivated; in the worst case, students may use their essays as an outlet to complain about the classroom process. This is exacerbated when a group of students responds to multiple prompts in a brief window, as in the fatigue point above.

5. Avoid scrubbing the data. It is natural for data scientists to remove atypical responses or early drafts, which can be seen as noise. Sometimes, district partners want to give only their best, exemplar responses from students. In that effort to "look good," they end up not supplying a complete set of essays or all the associated data. Districts sometimes use a prompt with multiple groups of students, but only provide essays that "fit" how they view successful responses to the prompt. By doing this, students at the low end of performance are intentionally omitted from representation. In practice, all sampled essays, including those in the tails of scoring, help in training. In general, a larger set of essays gives more for a model to learn from, leading to a model that can provide feedback to a broader range of students. Additionally, writing at different stages of completion is likely to appear within the live context of an AES intervention, and should not be totally foreign to the trained model. When logistically possible, collectors should collect early drafts of student work for scoring, to represent growth in essay quality over time. Filtering essays to remove outliers can be time-consuming and counter-productive.

6. Overcommunicate with partners. Many of the above

[3] Named entities are anonymized in public data releases and this excerpt, but the inaccuracy of this and other examples in the ASAP dataset has been confirmed through personal communication. For details on anonymization's impact on AES reliability, see (Shermis, Lottridge, and Mayfield 2015).

heuristics overlap. A well-documented plan for content collection will help all parties anticipate these issues and problem solve if factors could negatively impact the quality of the set. A collection process does not necessarily need to receive all data at one time; assigning batches of essays at typical peak writing periods during the school year yields higher-quality training sets.

Annotation Validity

Typically, scoring of large corpora of collected student text is not completed either by educators who collect those essays, or by AES practitioners. Instead, it is treated as a supervised annotation task and outsourced to a third party, consisting of large numbers of moderately trained participants and a smaller number of "lead scorers" with more experience and decision-making authority. Data is transferred to such a vendor after accounting for privacy regulations and removal of personally identifiable information. It takes work to build a relationship with a scoring vendor. If either side is not open to discussion and feedback, the partnership is not likely to meet the needs of both sides and will almost certainly not support a reliable partnership. This section specializes established best practices on corpus annotation[4] to the domain of rubric-based scores on student essays; in this context, the terms annotation and score are interchangeable.

1. Establish a collaborative process. Feedback and concerns from annotators are integrated into the scoring process. A vendor should read representative subsets of essays for scoring prior to large-scale annotation, and flag potential problems and requests for clarification. For instance, alignment between prompts and rubrics should be clear to vendors. If a set of essays does not match the expectations of the rubric, it should be identified upfront. Sometimes, this may be remedied with a clear rationale from the provider of collected data; in other cases, severe problems may lead to changes to a scoring rubric itself.

2. Identify anchor papers. In order to consistently apply a rubric to essays written to a specific prompt, an anchor paper review is crucial. In this process, scoring leads identify "anchor papers" that exemplify the score points across a rubric; these anchor papers are used to train the individual annotators. This process should be two-sided between researchers and the vendor; one group should submit their proposed set to the other group for consensus-building, to pair scoring expertise with knowledge of classroom context. This process should be iterative and anchor papers typically are added or removed through discussion prior to training annotators.

3. Develop clearly articulated rubrics. Clear lines should be drawn between performance levels in traits of a scoring rubric. Traits themselves must be distinguished from one another. Cross-correlated expectations across traits harm scoring quality. For example, the use of transition words may impact the overall quality of organization, and might also help to show the relationships between the claim and the evidence used to support it, but a rubric should be designed with each aspect of writing isolated to one trait. Annotators should be trained where students get "credit" for a particular skill. Additionally, rubrics must articulate a stepwise progression upwards through score points. The language that maps out what occurs within a trait has to be developed with key criteria for students and teachers, as well as alignment to accountability standards (as discussed above).

4. Build in a common vocabulary. Rubrics that map out score points often contain subjective phrases. Modifiers like "significant" or "thorough" can be interpreted differently by individual annotators. The distinction implied by these terms should be explained during training of annotators, and when reused, should have consistent meaning across traits. Descriptive language, particularly adverbs, should not vary in meaning across rubric traits or score points. Using anchor papers is a useful step in the process of defining these key modifier words, as they can be tied to authentic examples.

5. Use specific examples from student work. Essays collected authentically, following the Data Validity steps above, should be used in the collaboration on scoring best practices. Abstract ideas represented in a rubric should be rooted in real student writing to make them concrete. For example, "an objective tone" in a middle school essay collection is difficult to describe by adults, and may be easier to describe through examples of middle school text. Commentary by either researchers or vendors may be attached as qualitative explanations on student text during training, for rationale and clarity, but the concrete representation of concepts is more vital. Because there is no single correct way to construct an essay, multiple examples are often clarifying.

6. Avoid latent language ideologies. Student writing is produced in response to prompts that outline the language expectations for the assignment. In turn, annotators should score student responses based only on those explicit requirements. Annotators' personal preference or cultural familiarities may alter their holistic perceptions of writing quality. This can be expressed through subtle style biases, such as through dialect markers or grammaticality, or through hidden structural requirements like minimum word counts. Such subtle biases can disproportionately impact protected classes and students of color (Godley et al. 2006). This can undermine the validity of scoring, and it is therefore important to limit training of annotators to focus on the identified, specific writing requirements that were given to students during data collection.

7. Systematically evaluate scoring output. Consistently evaluate the scores given to each dataset before accepting scoring as complete. Design this evaluation system to capture the most common problems with calibration or misalignment of the scorers, and also the most common dataset flaws introduced by the data collection process. Scorer and data problems will likely be confounded, so an expert may need to determine the proper course of action in the case of poor results. The most commonly flagged error patterns include rare representation at the ends of a scale, extreme over-representation of a single score point, or poor agreement of individual annotators who are "out of sync" with the rest of a group. Unusually strong correlation with essay length, or cross-correlation between essay traits, is also a sign of rushed scoring.

[4] See for example (Hovy and Lavid 2010).

8. Share expectations around hiring and onboarding. Any vendor that works with essay scoring will have a standing process for selecting scorers, training, and calibration. This process is typically separate from, and in addition to, the scoring process for an individual dataset. For any organization working with a scoring vendor, it is essential that the organization has transparency into those established procedures. Beyond that, though, it is also important to make sure that both sides of the working relationship have a shared understanding of what processes are put in place to make sure that scoring leads are effectively chosen from a broader group of scorers, as they are the ones who must effectively train and disseminate critical information to the actual scorers. For valid scoring of training sets, there must be trust that scorers have experience and expertise in scoring the written work of adolescent students.

Evaluation in Practice

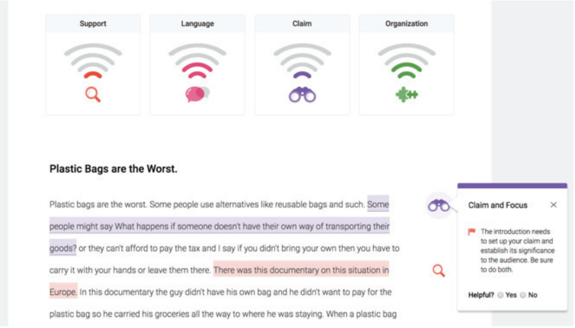
This remainder of this paper applies this framework to the evaluation of *Revision Assistant*, an AES intervention developed by Turnitin and primarily designed for formative classroom use and deployed at scale in American middle and high schools. *RA* emphasizes the importance of the writing process by reframing essay authorship as an on-going activity. The design of the system utilizes AES to embed an intensive revision process into student interactions with the system.

As students request automated scoring, feedback is also provided; *RA* highlights two relatively strong sentences and two relatively weak sentences (Woods et al. 2017). Instructional content appears alongside those sentences that helps students understands where they are excelling in their writing and where they should focus their revision efforts. Comments encourage students to take small, targeted steps toward iteratively improving their writing.

This design and pedagogical constraint is meant to provide students with the opportunity and the desire to engage in writing strategies around constant refinement and iteration. By creating an environment that directly connects student writing to feedback that encourages rework, it becomes clear to the student that good writing is the product of multiple drafts. The instantaneous nature of the feedback further aids students by creating an environment where revision can easily take place. Feedback cycles which could be days or weeks long are shorted to near-instantaneous feedback. This makes it significantly more motivating for students to revise and improve their work. The visual, game-like appeal of Wifi signals creates an atmosphere that encourages students to work and improve, without the feeling of finality from previous, summative AES systems.

Evaluating Curriculum Validity

Content within *RA* is wide-ranging (see Table 1). At time of this paper's authoring, content is distributed across genres, subject areas, and grade levels, though with more comprehensive in high school grade levels. Content is weighted towards English Language Arts. A subset of prompts has been specifically identified as appropriate for summative assessment purposes, while the rest are recommended for lower-stakes use only. This constitutes an appropriately diverse library with room for improvement in the physical sciences and in the younger grades.



Plastic Bags are the Worst.

Plastic bags are the worst. Some people use alternatives like reusable bags and such. Some people might say What happens if someone doesn't have their own way of transporting their goods? or they can't afford to pay the tax and I say if you didn't bring your own then you have to carry it with your hands or leave them there. There was this documentary on this situation in Europe. In this documentary the guy didn't have his own bag and he didn't want to pay for the plastic bag so he carried his groceries all the way to where he was staying. When a plastic bag

Claim and Focus ×

The introduction needs to set up your claim and establish its significance to the audience. Be sure to do both.

Helpful? ○ Yes ○ No

Figure 3: The user interface of *Revision Assistant*.

RA's scoring rubrics are genre- and grade-band dependent, but do not vary across prompts within those genres and grades. The rubrics are also designed for alignment with crosswalks to four different standards consortia - Smarter Balanced, PARCC, Texas Education Agency, and Florida Department of Education. Including all states which adopted the Common Core, this results in accountability crosswalks for 46 states and DC[5].

Source texts are weighted towards modern writing, with more than half of sources written in the 21st century. A wide range of identities are represented in source texts, including African American, Native American, Asian-American, and Hispanic authors, as well as texts by non-American authors. There is room for broader inclusion - fewer than 10% of texts are authored by women of color, and zero are written by nonbinary gendered authors.

Overall, *RA* provides strong curriculum validity for practitioners, including a wide-ranging library, alignment to standards in most school districts in America, and modern, diverse representation in authorship.

Evaluation Data Validity

For most prompts, collection of student work for *RA* training sets was timed across multiple months and in line with regular teaching practices.

Once tasks and rubrics were established with partners, best practices as described above were shared with district partners. For library expansion in the 2016-2017 school year, collections were timed in eight "waves" across 20 school districts, expanding the library by at least 50 prompts. Each wave consisted of between one and four participating school districts. Each wave consisted of pilot use of the prompt in classroom settings, evaluation of initial essays, and then a larger collection process across more classrooms.

Because of the nature of the work, waves were staggered, sometimes over a number of months. Though standards and pacing may provide guidelines for curricular materials, not all teachers assigned work on the same day, or even the same

[5] Education standards in Alaska, Nebraska, Oklahoma, and Virginia are not well-aligned to content in *Revision Assistant*.

Genre	Narrative Writing	11
	Informative Essays	31
	Text-based Argument	12
	Open-ended Argument	10
	Textual Analysis	17
Subject Area	English Language Arts	78
	Social Studies / History	32
	Physical Sciences	20
Grade Level	Grade 6	14
	Grade 7	22
	Grade 8	24
	Grades 9-10	32
	Grades 11-12	28
Summative Use	Timed High-Stakes OK	42
	Low-Stakes Only	42
Source Text Date	Before 1900	19
	1901-2000	33
	Since 2000	96
Expressed Identity of Source Author	Women (of color)	41 (10)
	Men (of color)	80 (16)
	Not presented	42

Table 1: Prompt library and source text distributions in *RA*.

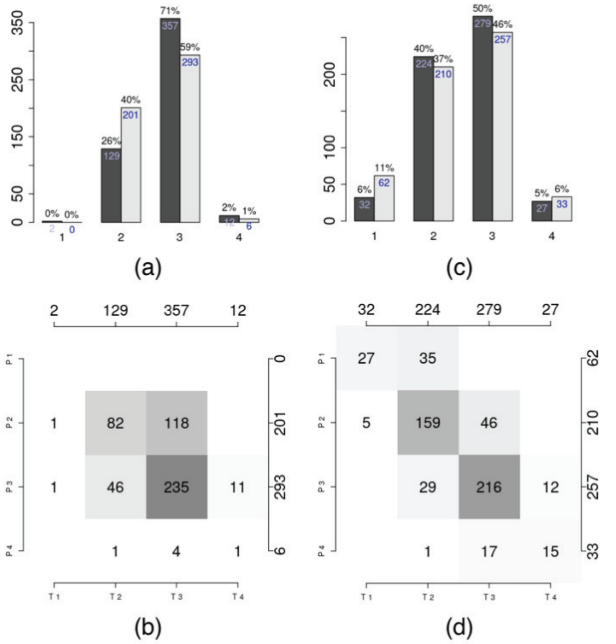


Figure 4: Analysis of a single prompt on a single trait before (a-b) and after (c-d) best practices for training set collection were put in place.

week. Each training set was collected by between 5 and 10 teachers, and waves consisted of a minimum of 5 and a maximum of 25 distinct prompts. Typically, groups of teachers within buildings were part of each wave, rather than working with individual teachers isolated from the process.

Datasets were vetted based on minimum training set size targets. Districts did not appear in waves as the sole participating district unless at least 500 unique student essays, spread across score points, could be reliably collected for each training set in that wave. This barrier prevented single-district training set collections in most cases. Instead, multiple districts share training set responsibility for each prompt, in order to ease the burden of collection on any one district. Essays were collected either through *RA* in an interface with no automated feedback, or were collected in other word processors and shared over secure file transfer.

Evaluating Annotation Validity

At a high level, performance of the model resulting from this collection process is measured through Quadratic Weighted Kappa, or QWK, the industry-standard method of evaluating model quality (Shermis and Burstein 2013). On this metric, which typically ranges from 0 to 1, industry best practices recommend performance of at least 0.6 before use even in low-stakes settings, and an optimal target of up to 0.8 for ”near-perfect” reproduction of expert scores. Further detail on model performance can be gleaned through evaluated score distributions across a training set's true and cross-validated predicted labels, and confusion matrices that highlight frequent mismatches between scores.

Figure 4 illustrates these visualizations in a model trained to score a 9th-10th grade essay prompt. Two sets of scored essays are shown, before and after application of the best practices above. Evaluation is completed through 10-fold cross-validation of a training set of 490 essays. Prior to implementation of best practices, the first model reaches only a QWK of 0.281, well below the industry benchmark. The final model's QWK reaches 0.749, above the threshold for high-stakes use. A more in-depth quantitative analysis is instructive and highlights the problems in hand-scoring when best practices are not followed. The distribution in the top left (a) presents counts of scores within the training set, both in ground truth (dark) and predicted (light) score sets. In the original hand scoring, 71% of all essays received the most common score of 3/4, and only 2 essays received the minimum score of 1/4. This “clumping” to the middle is common when oversight is minimal. The confusion matrix (b) highlights the challenge of automated scoring when essays are scored this way. The model learns to replicate observed scoring behavior, and ignores both the top and bottom of the scoring range altogether. Even within the two frequent score points, confusion is common; fewer than 65% of essays are scored exactly correctly, worse than would be expected from a trivial classifier that always predicted the majority class.

The right-hand column smooths out these problems somewhat. As seen in the score distribution (c), scores at the top and bottom of the scoring range now account for more than 10% of essays in the training set, enough for machine learning algorithms to identify reliable characteristics of 1/4 and 4/4 scores. The confusion matrix (d) shows that all four score points can now be reliably identified, even though the majority class still accounts for half of all essays. No errors greater than adjacent misses are made at any point during cross-validation. This pattern of improvement indicates a material improvement in scoring behavior as a result of the practices described in this paper.

Conclusion

Educators should expect AES to be held to a high standard when selecting interventions for use in classroom settings. Transparency in content selection, curriculum alignment, training set collection practices, school partnerships, and annotator hiring and training form a broad and comprehensive picture of automated essay scoring model behavior. This picture exceeds the transparency typical in the psychometric literature, which only gives sparing coverage to qualitative aspects of model training and emphasizes reliability.

Following best practices on all three categories - curriculum, data, and scoring - requires an extended partnership between school teachers, machine learning researchers, and annotators. This is a more interdisciplinary approach than statistics-driven arguments for validity, and requires more transparency than the AES community has previously been subjected to. Models trained at the end of a process that follows these best practices, however, both provide reliable scoring of student essays and support classroom instruction.

References

Attali, Y., and Burstein, J. 2004. Automated essay scoring with e-rater® v. 2.0. *ETS Research Report Series* (2).

Ball, D. L., and Cohen, D. 1996. Reform by the book: What isor might bethe role of curriculum materials in teacher learning and instructional reform? *Educational researcher* 25(9):6–14.

Burstein, J.; Chodorow, M.; and Leacock, C. 2004. Automated essay evaluation: The criterion online writing service. *AI Magazine* 25(3):27.

Crossley, S.; Varner, L.; Roscoe, R.; and McNamara, D. 2013. Using automated indices of cohesion to evaluate an intelligent tutoring system and an automated writing evaluation system. In *International Conference on Artificial Intelligence in Education*. Springer.

Foltz, P.; Hidalgo, P.; and Van Moere, A. 2014. Improving student writing through automated formative assessment: Practices and results. In *International Association for Educational Assessment Conference*.

Godley, A.; Sweetland, J.; Wheeler, R.; Minnici, A.; and Carpenter, B. 2006. Preparing teachers for dialectally diverse classrooms. *Educational Researcher* 35(8):30–37.

Grimes, D., and Warschauer, M. 2010. Utility in a fallible tool: A multi-site case study of automated writing evaluation. *Journal of Technology, Learning, and Assessment* 8(6).

Hovy, E., and Lavid, J. 2010. Towards a scienceof corpus annotation: a new methodological challenge for corpus linguistics. *International journal of translation* 22(1):13–36.

Keith, T. 2004. Validity of automated essay scoring systems. In Shermis, M., and Burstein, J., eds., *Automated essay scoring: A cross-disciplinary perspective*. Lawrence Erlbaum Associates. 266–290.

Markoff, J. 2013. Essay-grading software offers professors a break. *The New York Times* A1.

Mason, W., and Suri, S. 2012. Conducting behavioral research on amazons mechanical turk. *Behavior research methods* 44(1):1–23.

Nehm, R.; Ha, M.; and Mayfield, E. 2012. Transforming biology assessment with machine learning: automated scoring of written evolutionary explanations. *Journal of Science Education and Technology* 21(1):183–196.

Perelman, L. 2014. When "the state of the art" is counting words. *Assessing Writing* 21:104–111.

Riedel, E.; Dexter, S. L.; Scharber, C.; and Doering, A. 2006. Experimental evidence on the effectiveness of automated essay scoring in teacher education cases. *Journal of Educational Computing Research* 35(3):267–287.

Roscoe, R.; Allen, L.; Weston, J.; Crossley, S.; and McNamara, D. 2014. The writing pal intelligent tutoring system: Usability testing and development. *Computers and Composition* 34:39–59.

Sculley, D.; Holt, G.; Golovin, D.; Davydov, E.; Phillips, T.; Ebner, D.; Chaudhary, V.; Young, M.; Crespo, J.-F.; and Dennison, D. 2015. Hidden technical debt in machine learning systems. In *Advances in Neural Information Processing Systems*.

Shermis, M., and Burstein, J. 2013. *Handbook of automated essay evaluation: Current applications and new directions*. Routledge.

Shermis, M.; Lottridge, S.; and Mayfield, E. 2015. The impact of anonymization for automated essay scoring. *Journal of Educational Measurement* 52(4):419–436.

Shermis, M. D. 2014. State-of-the-art automated essay scoring: Competition, results, and future directions from a united states demonstration. *Assessing Writing* 20:53–76.

Socher, R.; Pennington, J.; Huang, E.; Ng, A.; and Manning, C. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of Empirical Methods in Natural Language Processing*.

Wilson, J., and Czik, A. 2016. Automated essay evaluation software in english language arts classrooms: Effects on teacher feedback, student motivation, and writing quality. *Computers & Education* 100:94–109.

Woods, B.; Adamson, D.; Miel, S.; and Mayfield, E. 2017. Formative essay feedback using predictive scoring models. In *Proceedings of the ACM Conference on Knowledge Discovery and Data Mining*.

The Potential Social Impact of the Artificial Intelligence Divide

Andrew B. Williams
Humanoid Engineering & Intelligent Robotics Lab
University of Kansas
andrew.williams@ku.edu

Abstract

This article describes the artificial intelligence (AI) divide, its social impact, and begins to prescribe policies to close this gap between those who benefit from AI data, algorithms, and hardware and those who are primarily exploited by them. Without a digitally aware, algorithm-literate public and an equitable public policy on AI, the AI divide will increasingly impact negatively those in lower socioeconomic classes in the U.S. and around the world.

Introduction

In the U.S. and other parts of the globe, the artificial intelligence (AI) divide threatens to make the lives of the poor, and perhaps the middle class, less healthy, prosperous, and safe. This growing AI divide finds its roots in poverty, discrimination, and joblessness. The AI divide represents the inequities between marginalized communities and adequately resourced communities caused by differing access to data, algorithms, and hardware used to power AI designed to promote the health, prosperity, and safety of privileged groups. These AI engines are increasingly only for those who can discern and grasp their benefits, afford them, and purchase the hardware that enable them.

The Ubiquity of AI

This paper does not dispute that there are many benefits of AI but rather focuses on its subtle and potentially dangerous impact on society. AI continues to slowly creep its way into our daily lives. What is interesting to note, that many things that are common place and not regarded as AI, were previously considered AI. For example, the object-oriented programming paradigm finds its roots in knowledge-based AI beginning with its predecessor, frame-based reasoning. Object oriented properties of inheritance and polymorphism taught in the 1990's were once part of artificial intelligence reasoning research.

Just in the last 10 years since the iPhone was created, we have seen what was considered one of the hallmarks of true artificial intelligence, natural language recognition and understanding, become increasingly more common and effective. Siri and Google Assistant, for example, continue to improve as these AI assistants combine big data and machine learning to better process and understand human speech. The eerie encroachment of AI into the home through Google Home and Amazon Alexa, shows the danger of society becoming too comfortable with AI. None of us would ever consider having a stranger living in our home and listening to our personal private conversations, yet this is what we do when we allow our smartphones or digital assistance, such as Alexa, to sit by our bedside constantly listening for us to say, "Hey Google." Recently, tech blogger reported that Google Home Mini was recording audio clips in his home and sending them to Google servers without being authorized and triggered by the "Hey Google" phrase (Burke, 2017).

The AI Divide

During the first dot com boom, minorities in the U.S. found themselves increasingly without the same internet connectivity and computer education. Even today, there is a divide in the number of lower socioeconomic schools with high speed or a one-to-one digital device policy (e.g. each child in the classroom being assigned an iPad) (Servon 2008). AI, which is often invisible to the human user in how it is developed, what input data is used, or how the algorithm works, can be used to benefit the few who created it or can pay for its benefits, and may biased against those it is designed to exploit (O'Neil 2016). The Google AI VP recently wrote that the immediate threat to humanity is not killer robots but rather the threat of biased algorithms (Knight 2017). He cites the example of an AI program that determines sentences for felons and has been shown to be biased against African Americans and Hispanic felons (Knight 2017).

Initially during the Microsoft Kinect infancy, the motion tracking and computer vision algorithms did not appear to work as effectively on darker skinned people as lighter skinned people (Ionescu 2010). This dilemma illustrated the concern of creating AI hardware and algorithms that are trained on only a small percentage of the worldwide population without regard to the range of the populations diversity. Hypothetically, autonomous driving algorithms and sensors trained only on people with lighter complexions and clothes that cover fewer parts of the body could pose a problem. The results could be disastrous if someone with darker complexions and wearing dark clothing covering most of the body were not detected by the car, which could result in a fatal collision with a pedestrian.

The AI divide is increasingly being used to influence thought and decisions. Twitter bots may have been used in the most recent U.S. presidential elections to influence the results (Bessi and Ferrara 2016). In a recent Stanford University study, middle school, high school, and college students in the U.S. were unable to determine when a digital article was authentic news or fake news (Wineburg et al. 2016). Amazon has been an early pioneer in using AI to learn user preferences and to predict what people will buy at a particular price. The AI-illiterate consumer will not know how to combat the exploitive nature of AI on their buying habits. AI will cause this consumer to spend more money for the same products as the more AI savvy consumer.

What if AI algorithms are used to predict the genetic causes of particular disease but trained only using data from a high socio-economic demographic? This AI program will be able to predict and maybe help preempt the disease through gene therapy for that demography. However, it would not effectively predict and prevent disease in lower socioeconomic class citizens. In the U.S. this group contains a relatively high number of African Americans and Latinos/Latinas. In this example, AI threatens the health of those in a lower socio-economic class.

Equitable AI Policy and Education

Our U.S. society is well passed the Industrial Age and deep into the Information Age. We have re-entered the AI Age. However, most of our citizens do not know how to write or read computer programs that are used to develop AI algorithms and systems. Some argue that learning how to code is as important as learning to read and write in one's primary language. If we are to close the AI divide, all kindergarten through high school students must learn how to read and write algorithms and code in a computer language. In addition, rudimentary graph theory would lay the ground work for students to learn more complex data structures and even machine learning algorithms, such as artificial neural networks.

Our society must erect structures and laws that protect the human dignity of individuals to insure all segments of our population are considered when developing health-related AI algorithms and systems that may eliminate previously human performed jobs. Recently in a discussion of the global ramifications of smart robotics on work and social justice, it was suggested by one economics expert that saving a few cents on a Big Mac through automation would be worth the loss of a job by a high school educated minority. A person's dignity is closely tied to their ability to work and contribute to society and their family's well-being.

The question of how lower socio-economic, marginalized communities can prosper in the AI age remains to be answered. AI combined with robotics for autonomous systems will pose an even more powerful threat to humans as even low-level service jobs such as janitorial or food service workers can be eliminated by AI-powered machines.

In the U.S., computer-related jobs in information technology and software continue to have job projections in the hundreds of thousands and perhaps millions in the foreseeable future. If there is shortage of people that can write code for industry, there will continue to be a shortage of citizens that can write, understand, and protect against AI-based exploitation of consumers and citizens across all socio-economic classes.

Conclusion and Future Work

The AI divide in the AI age threatens to create a society with even bigger wage gaps, more substantial joblessness, and lower health for marginalized populations. We must hold our governments accountable for investing in code and algorithmic literacy, policies that protect humans from detrimental AI exploitation, and create health access and predictive preventive care for all.

Acknowledgments

This work is partially funded by the Charles E. and Mary Jane Spahr endowment at the University of Kansas School of Engineering. The author wishes to acknowledge prior support from the GEM Fellowship and the NSF National Robotics Initiative.

References

Bessi, A. and Ferrara, E. 2016. Social Bots Distort the 2016 US Presidential Election, *First Monday*, Vol 21, No. 11, Nov. 7.

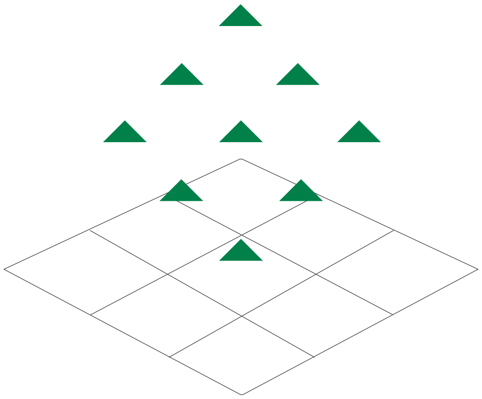
Ionescu, D. 2010, Is Microsoft's Kinect Racist? *PC World*, Nov. 4. Web. Jan. 28, 2018.

Knight, W. 2017. Google's AI Chief Says Forget Elon Musk's Killer Robots, and Worry about Bias in AI Systems Instead, *MIT Technology Review*, Oct. 3, 2017. Web. Jan. 28, 2018.

O'Neil, C. 2016. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy.* Crown Publishing, New York.

Servon, Lisa J. 2008. *Bridging the Digital Divide: Technology, Community and Public Policy*. Hoboken: John Wiley & Sons, Ltd.

Wineburg, S., McGrew, S., Breakstone, J., and Ortega, T. 2016. *Evaluating Information: The Cornerstone of Civic Online Reasoning.* Stanford Digital Repository.



Artificial Intelligence for the Internet of Everything

Compositional Models for the Internet of Everything

Spencer Breiner,[1,*] **Ram D. Sriram,**[1] **Eswaran Subrahmanian**[1,2]
[1]National Institute of Standards and Technology
Information Technology Lab
Gaithersburg, MD 20899, USA

[2]Carnegie Mellon University
Institute for Complex Engineered Systems
Pittsburgh, PA 15213, USA

Abstract

In this note we identify four fundamental characteristics of the IoE which are vexing to handle in practice: heterogeneity, composition, perspective and joint cognition. We discuss the way that each of these introduces a new dimension of complexity for the application of machine learning and artificial intelligence in the IoE. Finally, we introduce some mathematical methods from category theory which we believe can help to address these obstacles.

The Internet of Everything (IoE) represents the extension of computation into every facet of life, from recording a child's first steps to maintaining an elderly heartbeat (Sriram 2015). Every process in every domain, from soil aeration on a farm to search & rescue after a flood, will need to be reconsidered in light of new capabilities and efficiencies. Thus IoE applications are not only complex, but complex along many dimensions: they require different components connected by different communication schemes arranged in different patterns to satisfy different human needs. All of these exchanges must be organized based on rich, domain-specific semantic understanding in order to help rather than hinder these processes.

The fundamental characteristics of all of these applications is that they involve connected components that interact with one another through some more-or-less structured organization. Here component *must* be understood broadly to include (i) humans (as both subjects and actors) and computational resources, as well as (ii) the more traditional sensors and actuators, and (iii) the channels of interaction include energy, mass and information. Both the components and their organization must be chosen relative to a specific context, with specific goals and trade-offs between flexibility, robustness, resilience and efficiency.

In compositional systems like these, there is no escape from the issue of interdependence, where system behavior is generated by a complex interaction of component behaviors. After all, the whole point is that by arranging our system well we can achieve some benefit that couldn't be realized by the components in isolation! Interdependence is neither good nor bad, it is merely complex, though that complexity may obscure failure modes that were easier to discover in simpler systems. When those failures concern cardiology or search & rescue, that *is* bad.

We believe that formal methods from a branch of mathematics called category theory (CT) can help to overcome some of these difficulties. A flurry of recent work including (Baez and Fong 2015; Coecke and Kissinger 2017; Fong, Sobociński, and Rapisarda 2016; Vagner, Spivak, and Lerman 2015) indicates that string diagrams, a graphical syntax derived from CT, provide a formal foundation for the study of open and interconnected systems. These diagrams provide the representations needed to understand the interdependencies of the IoE, and suggest some possible tools for validating such systems based on deep connections with physics and computer science. This provides a strong *prima facie* argument that a CT-based approach could benefit the design and engineering of IoE systems.

Characteristics of IOE systems

In this section we review some core characteristics of IoE systems, and consider how composition interacts with these features.

Heterogeneity of components: The elements of an IoE system include, at a minimum, human actors and subjects, connected devices and cloud services. This indicates that to understand, predict or diagnose the behavior of an IoE system we may need to explore psychology, probability, dynamics and logic. Moreover, each element has its own logic of interaction. Sensors can support many subscribers whereas most actuators allow only one operator (at a time); humans are unpredictable in ways both good and bad. There are also other, less obvious components in our systems such as logical resources like encryption keys and personal data, which must be regarded as components of our systems if we hope to enforce information security in these systems.

Open interaction: The central feature of the IoE, in contrast to previous technologies, is that its components are expected to interact, and through that interaction unlock value and efficiency. More specifically, IoE components provide interfaces, both physical and logical, which may be coupled into a wide variety of different arrangements. Thus, to understand the behavior of an IoE system it is not enough to describe its components; we must also specify the architecture that wires those components together. In contrast to traditionally engineered systems, IoE systems will often be

[*] Corresponding author: spencer.breiner@nist.gov

provisioned on an ad hoc basis, sharpening the need for predictive tools for system behavior and security.

Multiplicity of perspectives: There is a tremendous range of viewpoints from which we may wish to consider an IoE system. Some of these are based on scale; an IoE system may range from a single individual (personal devices) to a building (HVAC) to a city or region (Smart Grid). This system of systems aspect of the IoE means that its local behavior may depend on any of these levels. The law introduces a new set of perspectives, including safety or privacy requirements as well as reporting for regulatory oversight. Economically speaking, the user of a component may not be its owner, and these two actors may be connected through a third-party platform.

Joint Cognition: Humans are components of IoE systems, but we are obviously unique in our capabilities, and the roles that we play in IoE systems will include sensing, actuation and control as well as subject of inquiry. When we wish to design systems in which human actors are components, our representations must go beyond artifacts, to include models of human behavior. When humans (or other autonomous agents) participate in the control loop for a complex system, it is essential that we know which information should be suppressed, what should be shared, and how that information should be presented in context. Compared to machines, humans are slow and error prone, but without our flexibility and global understanding systems become brittle and liable to fail.

Learning the IoE

Perhaps unsurprisingly, applications of artificial intelligence (AI) in the IoE are just like the IoE itself: heterogeneous and interdependent. When computation reaches into every facet of life it touches on all the types of learning that humans do and more beside. Thus what is needed, perhaps more than anything, to apply AI to the IoE is a framework to structure all of these potential applications.

The bread and butter of contemporary AI is the automation of specific information processing tasks, such as image classification or voice transcription. There are already many successful applications of such methods, and these will only continue to improve with new methods and more powerful devices.

However, the application of these methods in the IoE is still relatively inflexible. Training data for the problem must be collected and wrangled into a form appropriate for AI algorithms. The plumbing that connects AI to applications is usually done by hand on an ad hoc basis. Thus, what is required here is not new learning methods *per se*, but rather methods for more easily and efficiently specifying learning problems and integrating their results.

The open architecture of the IoE introduces an entirely different application of learning, concerning the design of IoE systems. Given an infrastructure of IoE devices, data, services and human & organizational actors, how can we achieve a stated goal within specified constraints? The size and diversity of the IoE ecosystem will ensure that humans cannot easily design such systems, especially given that many systems will be designed for one-off uses with on-the-fly provisioning.

The question that plagues these design processes is the ability to decompose and recompose these representations symbolically. Today, problems of (de)composition are usually addressed manually, completely outside the scope of formal models. Lacking explicit semantics, these ad hoc data interfaces are brittle and must be revised to handle even minor changes, leading to errors and unnecessary overhead.

This means that we will require, at a minimum, substantial artificial assistance in IoE system design. Supporting such applications will require, first of all, better representations for (potential) system designs, to serve as a concrete search space for the IoE design problem. Moreover, we must be able to link these architectures to rich semantic representations, so that the design system has access to the capabilities of the many devices available and an understanding of the goals presented by a user.

The IoE's variety of perspectives introduces more layers of complexity for learning. Given the multiplicity of scales, the parameters of one learning problem may be determined as the output of another. These interactions go both ways, with top-down modifications to operating parameters at lower levels (e.g., peak use incentives for electicity consumption) and bottom-up prediction for aggregate systems (e.g., monitoring expected consumption). Temporal perspective is also important, as system models will need to be updated as parameters change and components are replaced. This makes evolvability a crucial consideration for IoE design and engineering. Legal and economic perspectives introduce their own issues, requiring new ways of building constraints into learning problems in order to ensure that our systems meet their social obligations.

Perhaps the thorniest problem for learning in the IoE concerns joint cognitive systems. Today we do not trust machines to handle many of the tasks envisioned for the IoE. We can only build this trust incrementally, handing off some tasks from human to machine, integrating the two for others. To speed up this progression, we must engineer systems wherein machines can observe and interact with humans on line in order to better understand all the roles that we play.

Compositional Architectures

We have discussed some new and fundamental features which will be found in the IoE, and the way that these characteristics interact with machine learning and artifical intelligence. A central theme in this discussion is the need for new approaches to modeling complex systems, in order to account for these features. We believe that an approach based on the formal mathematics of *category theory* (CT) can help to address these new challenges.

CT is literally the mathematical study of compositional systems (Awodey 2010; Spivak 2014). The central feature of a category is a composition operation which allows us to combine two directed relationships $f : A \to B$ and $g : B \to C$ into a new relationship $f.g : A \to C$. Often, we think of f and g as resource-sensitive processes, and $f.g$ is the process which matches the output of f to the input of g.

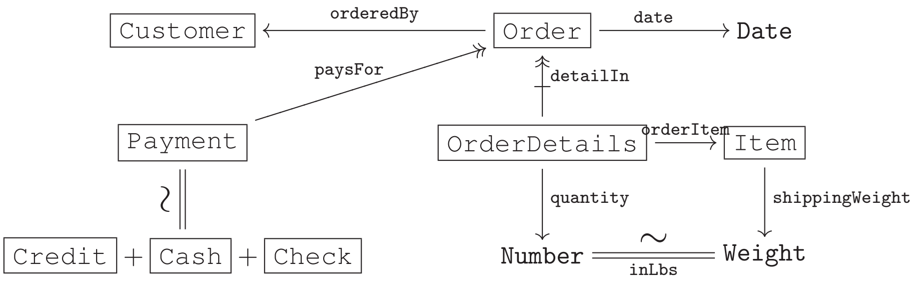


Figure 1: A categorical model in the style of a UML class diagram.(Breiner et al. 2017)

This is an exceedingly abstract perspective, which can be an obstacle for newcomers to the field, but this generality is necessary, as it allows for rich connections to formal methods in mathematics, physics and computer science. To see how the basic vocabulary of CT can be specialized to a variety of specific domains, see table 1.

CT can already provide well-understood connections with many of these domains. Indeed, it is a *lingua franca* allowing us to treat the range of these subjects using the same set of constructions, based on and extended from the basic vocabulary of objects and arrows. For the IoE, this means that CT can provide a suitable modeling formalism to capture the breadth of its heterogeneous components. Furthermore, connections between CT and formal logic mean that we can think of certain categorical models as logical theories, providing a powerful and expressive approach to knowledge representation which subsumes both database structures and ontologies (Spivak 2014).

CT also provides a candidate representation for compositional architectures, called *string diagrams* (Figure 2). A string diagram specifies a resource-sensitive functional decomposition of a complex process, describing the way that sub-processes feed resources among themselves in order to assemble the larger overall process. With origins in quantum computing (Penrose 1971), a recent flurry of research has produced applications of string diagrams ranging from electrical engineering (Baez and Fong 2015) to natural language processing (Coecke, Sadrzadeh, and Clark 2010).

A third crucial characteristic of CT is self-referentiality. We can think of categories $\mathbb{C}$ and $\mathbb{D}$ themselves as (informational) resources, and these can be linked together by directed relationships called functors $\mathbb{C} \to \mathbb{D}$. Functors provide translations between different information representations, providing concrete instructions for converting data expressed in terms of $\mathbb{C}$ into data expressed in terms of $\mathbb{D}$. These constructions allow us to bridge different information models, providing the means to manage and integrate the many perspectives found in the IoE.

In ...	the objects are ...	and the arrows are ...
Programming	Datatypes	Computable Functions
Physics	Configurations	Dynamical Evolution
Databases	Tables	Foreign Keys
Logic	Propositions	Proofs
Probability	Probability spaces	Stochastic Kernels
Data Science	Vector spaces	Matrices

Table 1: Interpretations of categorical language in a variety of domains.

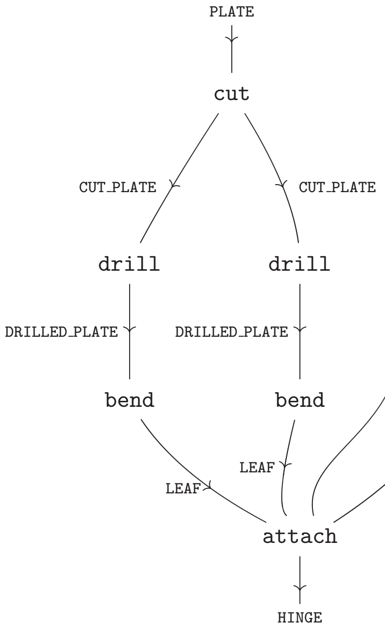


Figure 2: A string diagram model for a hinge-making process. (Breiner, Subrahmanian, and Jones 2017)

Sometimes these transformations will be bidirectional, providing a dictionary between one and the other; this could already be quite useful for data wrangling. More interesting, though, are cases in which the transformation cannot be reversed. In (Breiner, Subrahmanian, and Jones 2017) the authors showed that functors can be used to relate architectures at different levels of abstraction, so that $\mathbb{D}$ gives a functional refinement of $\mathbb{C}$. This allowed us to give a unified approach to process modeling from the production line to the factory to the global supply chain. In other cases, $\mathbb{C}$ might contain additional information which must be projected out in the passage to $\mathbb{D}$. This might be the case, for example, if $\mathbb{D}$ contains a simple process model that is extended in $\mathbb{C}$ to include security concerns by explicitly representing resources like encryption keys.

Finally, we note that the burden of joint cognition is mitigated somewhat by the diagrammatic character of CT. Unlike most formal disciplines, CT uses diagrams extensively as tools for simplifying complex arguments. We have already mentioned string diagrams, a formal syntax for process representation which is powerful enough to support calculations in quantum mechanics but can be read as easily as a flowchart. The semantic representations mentioned above can be presented through box-and-arrow diagrams which are not much different from UML class diagrams (Breiner et al. 2017). Some simple examples are shown in figures 2 and 1. These graphical representations support the way that humans think and understand, without sacrificing the formal mathematical character which is needed for machine interaction.

Conclusion

In this short note we have identified four critical characteristics of the IoE: heterogeneity, composition, perspective and joint cognition. Each of these introduces new challenges in the design and engineering of IoE systems, and this is reflected in the learning tasks which confront us.

Furthermore, we have suggested some reasons to think that a formal mathematical approach based on category theory can help us to address these challenges. These include deep connections with other formal methods, structured representations for compositional systems, structured mappings relating these different representations, and a graphical approach which supports human-machine interaction.

Category theory is generally regarded as an abstract area of pure mathematics, but in recent years the field of *applied* category theory has begun to grow. This area offers a wealth of potential applications to help tame the complexity of the IoE.

Disclaimer

Commercial products are identified in this article to adequately specify the material. This does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply the materials identified are necessarily the best available for the purpose.

References

Awodey, S. 2010. *Category theory*. Oxford University Press.

Baez, J. C., and Fong, B. 2015. A compositional framework for passive linear networks. *arXiv preprint arXiv:1504.05625*.

Breiner, S.; Padi, S.; Subrahmanian, E.; and Sriram, R. 2017. Deconstructing uml, part 1: The class diagram. Available by request: spencer.breiner@nist.gov.

Breiner, S.; Subrahmanian, E.; and Jones, A. 2017. Categorical models for process planning. Available by request: spencer.breiner@nist.gov.

Coecke, B., and Kissinger, A. 2017. *Picturing quantum processes*. Cambridge University Press.

Coecke, B.; Sadrzadeh, M.; and Clark, S. 2010. Mathematical foundations for a compositional distributional model of meaning. *arXiv preprint arXiv:1003.4394*.

Fong, B.; Sobociński, P.; and Rapisarda, P. 2016. A categorical approach to open and interconnected dynamical systems. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, 495–504. ACM.

Penrose, R. 1971. Applications of negative dimensional tensors. *Combinatorial mathematics and its applications* 221244.

Spivak, D. I. 2014. *Category theory for the sciences*. MIT Press.

Sriram, R. D. 2015. Smart networked systems and societies: A research agenda. https://www.youtube.com/watch?v=ywjekHO1pAM.

Vagner, D.; Spivak, D. I.; and Lerman, E. 2015. Algebras of open dynamical systems on the operad of wiring diagrams. *Theory and Applications of Categories* 30(51):1793–1822.

Internet of Things: Securing the Identity by Analyzing Ecosystem Models of Devices and Organizations

Kai Chih Chang, Razieh Nokhbeh Zaeem, K. Suzanne Barber
Center for Identity
The University of Texas at Austin
{kaichih, razieh, sbarber}@identity.utexas.edu

Abstract

The Internet of Things has become an integral part of our daily life. Its combination of network and emerging technology interlaced with each other results in a complicated environment that is left to us to understand and interact with. Information travels in the cyber world, not only bringing us convenience and prosperity but also jeopardy. Protecting this information has been an issue and commonly discussed in recent years. One type of this information is Personally Identifiable Information (PII), often used to perform personal authentication. With total cost of more than $40 billion since 2006, several reports of theft and fraudulent use of PII have been released. An all-embracing technique and system is needed in order to protect users from identity theft. In this paper, we present the Identity Ecosystem, a comprehensive identity framework that contains a mathematical representation of a model of Personally Identifiable Information attributes for people, and two novel models, devices and organizations, that have strong connections with the PII model of people. This research aims to combine the above three models and leads to better prevention against identity theft and fraudsters.

Introduction

The Internet of Things (IoT), the network of physical devices and the network connectivity that enables these devices to collect and exchange data, has been a growing paradigm in recent years. Undoubtedly, the main advantage of the idea of the Internet of Things is that it will have a significant impact on several aspects of the user's daily life and behavior.

For normal users, the most obvious effects of the IoT will be visible in fields like automation, e-health, and enhanced living quality, to name only a few. Since the concept of "Smart City" has been commonly discussed in recent years, it is without doubt the new paradigm that will play an influential role in the near future. Similarly, from the perspective of business users, the most apparent consequences will be equally visible in fields such as automation and industrial manufacturing, logistics, business/process management, and intelligent transportation of people and goods (Atzori, Iera, and Morabito 2010). The IoT's influence is growing, while some potential problems are gradually surfacing.

Data protection has been a problem since the network began to evolve. With the commercialization of the Internet, security issues have been extended to cover personal privacy, financial transactions and cyber-theft threats. In the paradigm of the Internet of Things, security and safety are inseparable. Whether it is accidental or malicious, interfering with personal mobile phones, hacking into the computers of an organization, and other similar acts pose a threat to human privacy, property, and even life. Even arbitrary data, like a temperature, might be related to a user when it is combined with other data like location or is profiled over a period of time. Privacy becomes crucial in the Internet of Things[1]. How to protect the privacy of individuals, that is, to safeguard this identity information to prevent identity theft, has become one of the mainstream topics discussed today. Federal Trade Commission (FTC) has estimated an annual loss of over 15 billion dollars from identity theft in 2006 (Synovate 2007). In 2010 this figure had more than doubled, as 8.1 million U.S. adults were the victims of identity theft or fraud, with total costs of $37 billion (Miceli and Vamosi 2011). Identity theft, according to the National Institute of Justice, has become the prime crime in the information age, with an estimated 9 million or more incidents each year (Newman and McNally 2005). Identity theft threatens our safety and property, unless we can truly prevent fraudsters from identity breaching.

In this paper, we seek to discuss the identity theft issues most relevant to people, businesses, and devices. The first aspect of identity is the one that identifies people, or Personally Identifiable Information (PII). The IoT world would benefit from this one kind of identifiers. However, there is no special identifier in the IoT world and there will never be one in the near future[1]. For example, public classic IP-addresses (IPv4 addresses) are a rare resource. Access providers use IP-address pools and "re-use" IP-addresses by dynamic assignment, which means that with every mobile phone login, the mobile client might be assigned to an IP-address different from the one that was assigned from last login (Friese, Heuer, and Kong 2014). Our understanding of this personally identifiable information is not enough.

[1] "Q&A Identity & Internet of Things", Ingo Friese, and Jeff Stollman, and Scott Shorter. http://kantarainitiative.org/confluence/display/IDoT/Q&A+Identity+&+Internet+of+Things (accessed Oct, 2017).

In the pursuit of security, this information needs to be understood and valued. Being merged with online attributes and offline attributes, the cyber world has been assimilated into people's everyday world. Online attributes are composed of one's social media accounts, online shopping patterns, passwords, email accounts and so on. Offline attributes are those related to the physical world such as bank accounts, credit and debit cards, social security number, fingerprint, blood type, etc. A more comprehensive online identity framework is needed based on a sound understanding of PII (Liang 2014).

The current Identity Ecosystem is limited to a single general model that hypothesizes only individuals have PII. But in fact, a mobile phone tracks its owner's current location. A laptop stores plenty of one's private information. Even one's sports watch or e-health equipment are transmitting his/her body status such as body temperature and heart rate. This information travels in the cyber world through the Internet. Eventually, it flows into the server of a company or an organization. A security incident at that organization may expose personal information that belongs to a large number of people and result in monetary loss. Taking the above scenario into consideration, in this paper, we introduce two extra models: devices and organizations. Only by combining the graphic model of people, devices, and organizations will we obtain comprehensive knowledge of the operation of PII in the cyber world.

In the following section, we briefly introduce how Ecosystem works, and then introduce our two models. Then we discuss our data resources. Finally, we present the conclusion and proposed future work.

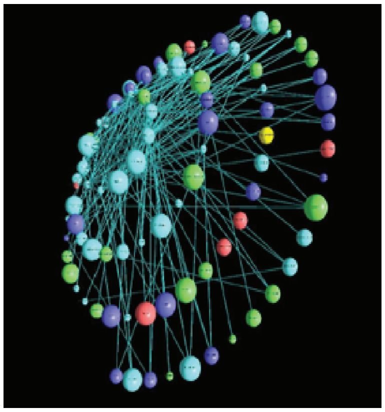
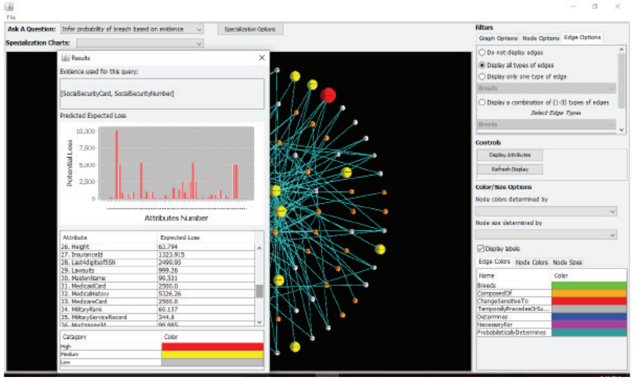


Figure 1: The 3D graphic model shown in Ecosystem.



Ask A Question: Infer probability of breach based on evidence

Specialization Charts:

Results

Evidence used for this query:

[SocialSecurityCard, SocialSecurityNumber]

Predicted Expected Loss

Attribute	Expected Loss
26. Height	63.794
27. InsuranceId	1323.915
28. Last4digitsofSSN	2490.95
29. Lawsuits	999.26
30. MaidenName	99.531
31. MedicaidCard	2500.0
32. MedicalHistory	5326.26
33. MedicareCard	2500.0
34. MilitaryRank	69.137
35. MilitaryServiceRecord	344.8
36. MortgageId	99.985

Category	Color
High	🟥
Medium	🟨
Low	▨

Filters

Graph Options | Node Options | Edge Options

○ Do not display edges
● Display all types of edges
○ Display only one type of edge
Breeds
○ Display a combination of (1-3) types of edges
Select Edge Types
Breeds

Controls

Color/Size Options

Node colors determined by

Node size determined by

☑ Display labels

Name	Color
Breeds	🟩
ComposedOf	🟧
ChangeSensitiveTo	🟥
TemporallyPrecedesOrSa...	▨
Determines	🟦
NecessaryFor	🟪
ProbabilisticallyDetermines	🟦

Figure 2: Asking Queries: Infer the Probability of Breach.

Ecosystem Models

As mentioned in section 1, the Identity Ecosystem developed at the Center for Identity at the University of Texas at Austin has constructed a graph-based model of people. It provides a statistical framework for understanding the value, risk and mutual relationships of personally identifiable information attributes. It uses a Network Model to simulate the relationships among PIIs for individuals. It allows predictions in the presence of interventions and it is able to handle incomplete data sets. It is visualized in a 3D graphic model and can be moved and rotated. The Ecosystem allows the users to choose a node property, such as value or risk, to determine node sizes and colors in the 3D graphic model (Nokhbeh Zaeem et al. 2016). Figure 1 shows the graph visualized in Ecosystem. Three interesting questions that Ecosystem can answer are inferring probability of breach based on evidence, detecting most probable origin of a breach, and finding breach hot-spots.

- Effect of exposure: Assuming a set of attributes is exposed, the Bayesian inference model of Ecosystem calculates the change in the probability of exposure of other attributes. The Ecosystem can also show the predicted expected loss of the set of attributes compromised. Figure 2 shows how the probability of breach for other attributes changes, once the Social Security Number and Social Security Card attributes have been breached. Multiple attributes can be selected as evidence at the same time. It also shows potential loss after such a breach scenario.
- Cause: If an individual finds out that his/her PII is compromised, the Ecosystem can help to detect the most probable origin of the breach through selecting identity information as the evidence and running the query.
- Cost/Liability: The Ecosystem can calculate attributes which have the highest cost (breach hot-spots) and should be best protected.

So far, Ecosystem can answer these questions for the model of people's PII. In this section, we introduce two novel models that have a strong connection with the PII graph of people: devices and organizations. In an IoT world there will exist a vast amount of raw data being continuously collected. It will be necessary to develop techniques that convert this raw data into usable knowledge (Stankovic 2014). The identity data would be one of these types of data. We define a person's identity as a set of information that are linked to the person. The identity data not only exist for people, but are also extended into our mobile phones, vehicles, online applications, and so on. Hence, it is important to build the concept of identity for our devices.

51 attributes of PII of devices			
Administrator	AdministratorPassword	AdministratorUserID	Application
ApplicationType	ApplicationVendor	BusType	Cache
CircuitDesign	Color	CookieWipe	GeolocationStreetAddress
GeolocationZipCode	InventoryTag	IPAddresses	MACAddress
ManufactureLocation	Manufacturer	MemorySize	MemoryType
ModelNo	NetworkCards	NetworkConnectionSpeed	NetworkConnectionType
NetworkProxySettings	NumberOfAssociateEmails	NumberOfPorts	NumberOfTransactions
OpenDeviceIdNumber	OperatingSystemType	OperatingSystemVendor	OrganizationalLocation
Owner	PortNumbers	PowerFrequency	PowerUsage
ProcessorType	RegistryProperties	Reputation	SerialNo
ShipDate	SwitchingRate	TimeZone	TransactionFlags
TransactionsByCountry	TransactionsProfile	TransactionVolCountry	TransactionVolumeTotal
UniqueDeviceIdentifier	Users	Watermark	

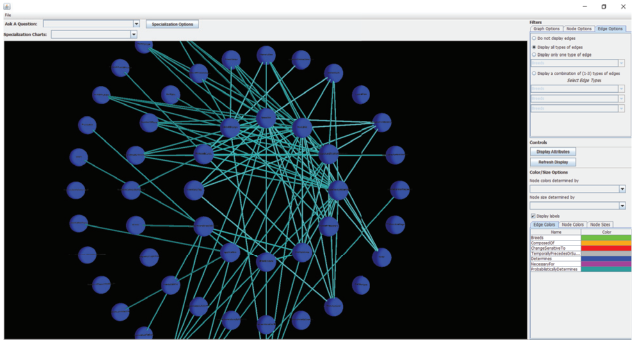
Table 1: List of all nodes of devices

Devices

Recently the concept of "Smart City" has rapidly risen (Dohler et al. 2011). Smart Cities consists of smart phones, mobile devices, sensors, embedded systems, smart environments, smart meters, and instrumentation sustaining the intelligence of cities (Schaffers et al. 2011). As a result, the relationship between people and devices has become blissfully tight. From mobile phones and laptops to GPS, sports watches and even to baby monitors, technical devices are collecting our PII anytime and anywhere.

We constructed a list of PII of items according to devices' characteristic, function, affordances and other documents (Gubbi et al. 2013) (see Table 1). Then we endeavored to manually find the links between these nodes. As a result, we generated a model graph of devices' PIIs. Figure 3 is a snapshot of the device graph presented by Ecosystem. Its main point lies in the links to the person's PII graph.

In fact, it it not uncommon to see the relationships between devices and people in our daily lives. The IP and MAC address and the vehicle's GPS imply one's location. Plenty of personal information have been stored in applications in one's mobile phone and computers. Moreover, sports and health devices are collecting one's body temperature and heart rates. Recent advances in mobile technology and cloud computing have inspired numerous designs of cloud-based health care services and devices. Within the cloud system, medical data can be collected and transmitted automatically to medical professionals from anywhere and feedback can be returned to patients through the network (Deshpande and Kulkarni 2017). This progress presages the growing convenience of collecting PII through devices, while it concerns many with respect to privacy of personal information.



Name	Color
Breeds	🟩
ComposedOf	🟧
ChangeSensitiveTo	🟥
TemporallyPrecedesOrSu...	⬜
Determines	🟦
NecessaryFor	🟪
ProbabilisticallyDetermines	🟦

Figure 3: The model of devices shown in Ecosystem.

Organizations

We are also interested in the relationship between people and organizations since activities that people trigger or

57 attributes of PII of organizations			
401KAdminitrator	AccessCards	Acquisitions	Address
Attorney	BalanceSheet	BankAccountNumber	BankingInstitution
Bankruptcy	BetterBusinessBureauRec	BoardOfDirectors	BusinessPropTaxNumber
BusinessType	Buyer	CAGENo	ComputerOrIPAddresses
CreditCardNumber	CreditCards	CreditRating	CreditScore
Customers	DateEstablished	DUNSNo	EmailAddress
Employees	FacebookAccount	FederalTaxID	IncorporationState
InStorePurchasingPatterns	Investors	JCPCertificationNumber	LawsuitRecords
License	LoanNumber	LoginPasswords	LoginUserId
LoyaltyCards	Name	Officers	OnlinePurchasingPatterns
OperatingSystem	Owner	Patents	PhoneNumber
PLStatement	PurchasingPatterns	SalesTaxNumber	SICCode
StockExchTickerSymbol	Stockholders	StockPrice	TwitterHashtag
VendorAddress	VendorName	VendorNumber	WebsiteURL
WorkforceCommissionID			

Table 2: List of all nodes of organizations

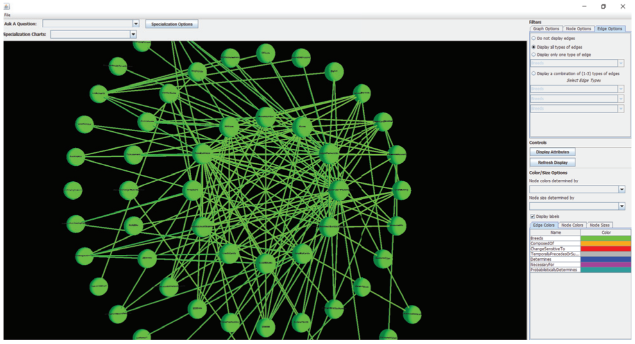


Figure 4: The model of organizations shown in Ecosystem.

be part of everyday are related to companies and organizations. Identity data breach through organization is now a widespread problem around the globe. A security incident at an organization may expose personal information that belongs to a large number of people. The goal here is to construct a graph-based model of organization PII attributes and analyze its linkages to the people PII graph in order to help the Identity Ecosystem's investigation.

The most fundamental PII of organizations is people (see Table 2). Employees, officers, supervisors, board of directors, and even CEOs are integral parts of the model. They have the ability to access most machines in the company or factory which store most customers' information. Hence, any information that is related to the machines would be treated as an organization attribute. We have also focused on documents that organizations would use in various situations by investigating the Certification of Formation from Texas Secretary of State[2]. Figure 4 is a snapshot of the organization graph presented by Ecosystem.

Community websites spread rapidly, not to mention the shopping websites. Every time one applies for a membership, he/she gives personal information to the organization that owns the website. Once the data has been received, the organization has the duty to keep these PIIs safe. However, breaches happen everywhere. Through the servers of an organization, customers' banking accounts could be exposed and misused by others. It is through these means

[2] "Texas Secretary of State", Rolando B. Pablos. https://www.sos.state.tx.us (accessed August, 2017).

that, answering the queries of Ecosystem helps our goal to thwart identity thieves and fraudsters.

Data Sources

Modeling Identity Attributes (Nodes)

The Ecosystem distinguishes various properties of identity attributes. Take attribute's type for instance; we divided the attribute's type for a person into four categories: What You Are, What You Have, What You Know, and What You Do. In our previous work (Nokhbeh Zaeem et al. 2016), we briefly introduced every property of identity attributes in detail. Here we only introduce the way we came up with nodes for devices and organization using this classification. We also refer to a list of documents from Texas Secretary of State[2] and (Gubbi et al. 2013) in our methodology.

What You Are For a person, it means a person's physical characteristics, such as fingerprints and retinas. For a device, it means the type of a device. It can be a laptop, a smart watch, a sensor, and so on. It is also related to a device's hardware configuration, such as circuit design and power usage. For an organization, it can also be its type. Also, it can be an organization's icon, such as stock market icon.

What You Have For a person, it means credentials and numbers assigned to the person by other entities. For a device, it can be its model numbers, serial numbers, and inventory tags. For an organization, it can be its sales tax number and DUNS number.

What You Know For a person, it means information known privately to the person, such as passwords. For a device and an organization, it means any information that is stored in them. So all information stored in an app or customer information stored in a server of an organization are all related to this type.

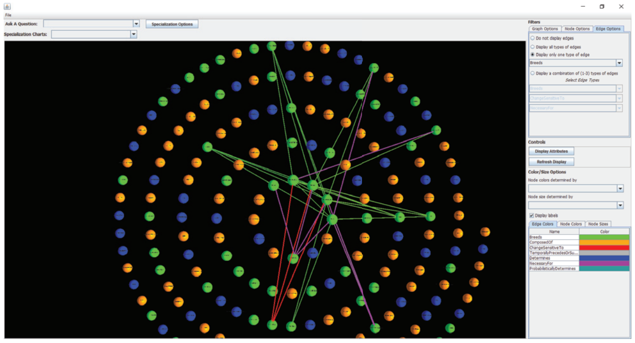
What You Do For a person, it means a person's behavior and action patterns, such as GPS location. What a device can do is often related to its application type, but for an organization of an online shopping website, it can be its online shopping pattern.

Modeling Identity Relationships (Edges)

The Ecosystem displays each attribute as a node. These nodes are related to each other in many different ways. The Ecosystem displays each relationship as an edge. We divided the type of relationships between a person's PII into 7 categories (Nokhbeh Zaeem et al. 2016). According to this classification, we are able to assign edges between nodes for devices and organizations.

Breeds α Breeds β means that an instance/value of α may be used in order to create a instance/value of β. For example, a driver's license breeds a boarding pass. A publication in an organization breeds its patent.

Composed Of α Composed Of β means that for any value α_i of the attribute α there is a value β_j of the attribute β such that β_j is a proper part of α_i. For example, a device's circuit design is composed of bus type and memory type.



○ Do not display edges
○ Display all types of edges
● Display only one type of edge
Breeds
○ Display a combination of (1-3) types of edges
Breeds
ChangeSensitiveTo
NecessaryFor
☑ Display labels

Name	Color
Breeds	🟩
ComposedOf	🟧
ChangeSensitiveTo	🟥
TemporallyPrecedesOrSu...	⬜
Determines	🟦
NecessaryFor	🟪
ProbabilisticallyDetermines	🟦

Figure 5: Three types of edge including Breeds, Changes Sensitive To, and Necessary For shown in Ecosystem.

Changes Sensitive To α Changes Sensitive To β means that for any person P with attributes α and β, if the value of β changes for P, then the value of α changes for P. For example, an organization's customers change with its reputation.

Temporally Precedes α Temporally Precedes β means that for any person P, P must possess some value of attribute α before P can possess a value of attribute β. For example, a mobile phone's login password precedes applications installed on it.

Determines α Determines β means that for any person P with attributes α and β, the value of α possessed by P implies the value of β possessed by P. For example, the IP address and MAC address of a server owned by an organization determines its geolocation.

Necessary For α Necessary For β means that for any person P, if P has a value for the attribute β, then P has a value for the attribute α. For example, the name of an organization is necessary for an employee's access card.

Probabilistically Determines α Probabilistically Determines β means that for any person P with attributes α and β, P having a given value of α implies that P probably has some particular value of β. For example, an organization's better business bureau record ratings represent how well the business is likely to interact with its customers, so it probabilistically determines its reputation.

By assigning edges between identity attributes, we generated our graphic-based models. The relationship between A and B is shown with a directed edge from A to B in the Ecosystem. The user can select to view one or multiple types of edges at a time. Different types of edges are shown in different colors in Ecosystem. Figure 5 shows a snapshot of displaying three types of edge in Ecosystem.

ITAP

To obtain accurate input data, we utilized the Identity Threat Assessment and Prediction (ITAP) project at the Center for Identity. ITAP is a risk assessment tool that increases fundamental understanding of identity theft processes and patterns of threats and vulnerabilities. A team of modelers at the Center for Identity analyzes identity theft news and stories on a daily basis to model the value of identity attributes and their risk of exposure. The ITAP database is large and continually growing, with approximately 5,000 incidents captured in the model to date.

Conclusion and Future work

In this paper two novel graphic-based models were introduced which offer an insight into how personally identifiable information is utilized within the cyber world. The model of devices and organizations imply the proliferation of technology as the Internet brings closer the vision of the Internet of Things. We are interested in the connections of these models to the PII model of people. Previously the Identity Ecosystem could answer three interesting questions, which were based only on the PII attributes of people. By combining and analyzing the people, device, and organization models together, we expect to derive more accurate and comprehensive results from the Identity Ecosystem. The potential structures and types of this cooperation framework and innovation resources from ITAP need further examination and interpretation since the Center for Identity envisions using low risk, low value, and high uniqueness PII for identifying and authenticating people in the future IoT-based society.

References

Atzori, L.; Iera, A.; and Morabito, G. 2010. The internet of things: A survey. *Computer Networks* 54(15):2787 – 2805.

Deshpande, U. U., and Kulkarni, M. A. 2017. Iot based real time ecg monitoring system using cypress wiced. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering* 6(2):710–720.

Dohler, M.; Vilajosana, I.; Vilajosana, X.; and Llosa, J. 2011. Smart cities: An action plan. In *Barcelona Smart Cities Congress*, 1–6.

Friese, I.; Heuer, J.; and Kong, N. 2014. Challenges from the identities of things. In *IEEE World Forum on Internet of Things (WF-IoT) 2014*, 1–4.

Gubbi, J.; Buyya, R.; Marusic, S.; and Palaniswami, M. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 29(7):1645 – 1660. Including Special sections: Cyber-enabled Distributed Computing for Ubiquitous Cloud and Network Services & Cloud Computing and Scientific Applications Big Data, Scalable Analytics, and Beyond.

Liang, Z. 2014. Specialization in the identity ecosystem. Master's thesis, The University of Texas at Austin.

Miceli, D., and Vamosi, R. 2011. 2011 2011 Identity Fraud Survey Report: Identity Fraud Decreases but Remaining Frauds Cost Consumers More Time & Money. Technical report, Retrieved from Javelin Strategy and Research: https://www.javelinstrategy.com/research.

Newman, G. R., and McNally, M. M. 2005. Identity theft literature review. Technical report, Retrieved from National Criminal Justice Reference Service: https://www.ncjrs.gov/App/Publications/abstract.aspx?ID=210459.

Nokhbeh Zaeem, R.; Budalakoti, S.; Suzanne Barber, K.; Rasheed, M.; and Bajaj, C. 2016. Predicting and explaining identity risk, exposure and cost using the ecosystem of identity attributes. In *IEEE International Carnahan Conference on Security Technology*, 1–8.

Schaffers, H.; Komninos, N.; Pallot, M.; Trousse, B.; Nilsson, M.; and Oliveira, A. 2011. *Smart Cities and the Future Internet: Towards Cooperation Frameworks for Open Innovation*. Berlin, Heidelberg: Springer Berlin Heidelberg. 431–446.

Stankovic, J. A. 2014. Research directions for the internet of things. *IEEE Internet of Things Journal* 1(1):3–9.

Synovate. 2007. 2006 identity theft survey report. Technical report, Retrieved from Federal Trade Commission: https://www.ftc.gov/sites/default/files/documents/reports/federal-trade-commission-2006-identity-theft-survey-report-prepared-commission-synovate/synovatereport.pdf.

Meta-Agents: Managing Dynamism in the Internet of Things (IoT) with Multi-Agent Networks

Hesham Fouad, Ira S. Moskowitz
Information Management and Decision Architectures Branch, Code 5580
Naval Research Laboratory
Washington, DC 20375

Abstract

In our talk we discuss Meta-agents frameworks for working within the Internet of Things (IoT) systems. In particular we discuss our own Meta-agent system called SENtry Agents (SAGE). We discuss why such systems must follow Simon's laws of the Artificial, and because of that must be Holonic.

Introduction

The rapid growth of the Internet of Things (IoT) (Columbus 2017) has created fertile ground for emerging research on a variety of existing and novel problems such as privacy, cyber security, big data, and self-adaptation/self-organization. A single IoT device (e.g. a thermostat) may serve a particular purpose, but the conglomeration of multiple devices to serve a human, or virtual entity's global objective, is the true promise of IoT. The vision of pervasive or ubiquitous computing is the existence of computational middleware that manages a set of IoT resources so that they constructively cooperate with each other to achieve the above global objective. We note that this global objective, or objective for short, can be as trivial assisting a homemaker in the shopping and preparation of family meals, or as important as a dynamic medical sensor network assisting in the care of hundreds, or thousands or patients.

An inherent challenge of IoT is that computational entities must operate in a highly dynamic environment, with emergent phenomena, that continuously change context, and do so in unpredictable ways. Existing software design paradigms cannot address such problems because they approach the bounds of complexity manageable by a human designer. In contrast, for the IoT situation, software paradigms need to be extended, and possibly totally rewritten to deal with the Artificially Intelligent (AI) situation brought about the rise of self-organizing, adaptive multi-agent systems (MAS) (Bernon et al. 2006), (Bernon et al. 2004),(Gardelli et al. 2006),(Gleizes et al. 2007).

In (Mihailescu, Spalazzese, and Davidsson 2017), Mihailescu et al. introduce the idea of Emergent Configurations (EC) for the IoT driven by user requirements. The idea is to dynamically orchestrate heterogeneous "things" in a manner that enables goal-directed behavior in support of a user's requirements. In our talk we will explore the idea of online agent creation and deployment as a way to realize EC. In the context of our work, Meta-agents are agents in a multi-agent software paradigm that utilize reasoning to both construct and deploy special purpose agents that form an EC. Unfortunately and opportunistically, reasoning models that can support the idea of meta-agents have not been explored. We assess the feasibility of using the prevalent Belief-Desire-Intention (BDI) reasoning (Georgeff et al. 1999) for modeling meta-agents. We further propose extensions to the model in support of meta-agents. Finally, we introduce the SENtry Agents (SAGE) multi-agent framework. SAGE is a novel multi-agent framework developed by us at NRL that supports meta-agents, online agent creation, as well as agent migration.

A necessity is that our Meta-agents obey Simon's laws of the Artificial (Simon 1990),(Valckenars, Brussell, and Holvet 2009). Simon's laws consist of three major tenants. First is *Bounded Rationality*. This states that our Meta-agent system attempts to make the best decisions based upon a limited amount of information. Real AI systems are not the Oracle of Delpi! They are very fast algorithms running with the best data possible at the time.

Secondly, our Meta-agent system exists in a *Demanding Environment*.

The third law concerns itself with a *Dynamic Environment*.

In our talk we will discuss these in more details and with examples. The bottom line is though that since our Meta-agent system follows Simon's laws of the Artificial it must in fact be a Holonic system. A system is Holonic (Valckenars, Brussell, and Holvet 2009) if it is designed on a top down pyramidal tree-like structure. It is analogous to a proof by induction in its set-up. That is we start off at the highest and first structure, then once we have that we move on to the second, then to the third, etc. Each level gets more and more complex, but builds on the levels before it. It is also similar to the idea of a mathematical filtration. We can stop at level n, but we do better at level n+1. That is our software gets more and more refined at each level. Again, in the talk we will discuss how we design SAGE in a Holonic manner.

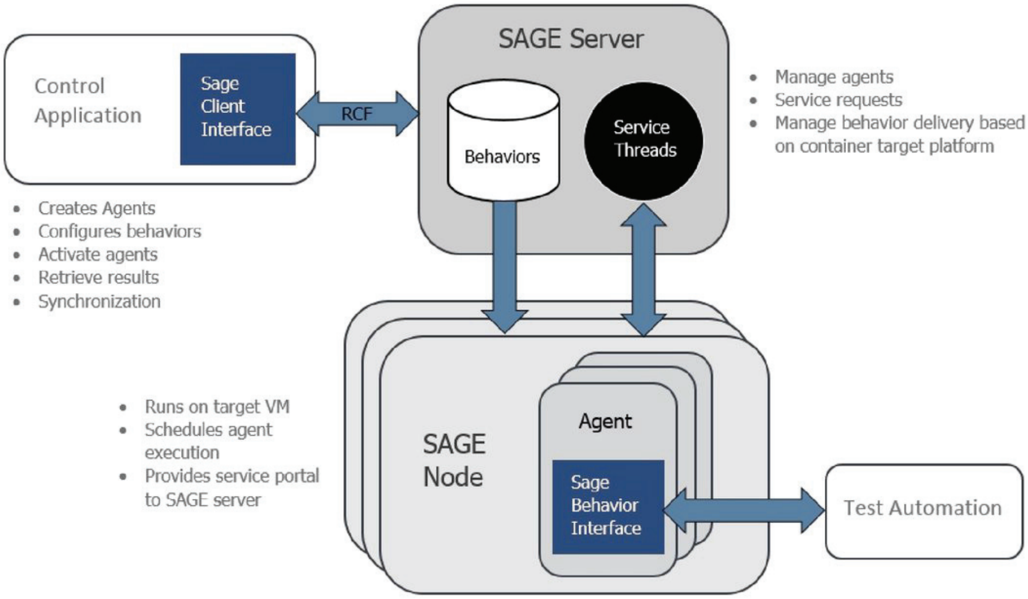


Figure 1: SAGE Framework form (Fouad et al. 2017).

SAGE

SAGE (Fouad et al. 2017) was initially developed to deal with the issues of agent generation in a service oriented architecture (SOA). The SOA envisioned was for DOD's Tactical Service Oriented Architecture (TSOA). SAGE is a multi-agent system written in C++. SAGE uses dynamic agents, by this we mean agents which can spawn and create other agents without being tied to specific actions. In out talk we will go into the details of the SAGE meta-agent framework.

Acknowledgements

We thank Bill Lawless and Antonio Gilliam for their assistance.

References

Bernon, C.; Camps, V.; Gleizes, M.; and et al. 2004. Tools for self-organizing applications engineering. In Serugendo, G. D. M., ed., *Ser. Lecture Notes in Artificial Intelligence*, volume 2977. Springer. 283–298.

Bernon, C.; Chevrier, V.; Hilaire, V.; and et al. 2006. Applications of self-organising multi-agent systems: An initial framework for compariso. *Informatica* (30):73–82.

Columbus, L. 2017. 2017 Roundup of internet of things forecasts.

Fouad, H.; Gilliam, A.; Guleyupoglu, S.; and Russell, S. M. 2017. Automated evaluation of service oriented architecture systems: a case study. In *Next-Generation Analyst V*, SPIE Defense + Security.

Gardelli, L.; Viroli, M.; Casadei, M.; and et al. 2006. Designing self-organising MAS environments: The collective sort case. In Weyns, D.; Parunak, H. V. D.; and Michel, F., eds., *Ser. Lecture Notes in Artificial Intelligence: Environments for Multi-Agent Systems III*, volume 4389. Springer. 254–271.

Georgeff, M.; Pell, B.; Pollack, M.; Tambe, M.; and Wooldridge, M. 1999. The belief-desire-intention model of agency. In *Proceedings of the 5th International Workshop on Intelligent Agents V: Agent Theories, Architectures, and Languages (ATAL-98*, 1–10.

Gleizes, M.; Camps, V.; Georgé, J.; and et al. 2007. Engineering systems which generate emergent functionalities. In *Proc. Int. Workshop on Engineering Environment-Mediated Multi-Agent Systems (EEMMAS 2007)*, 58–75.

Mihailescu, R.; Spalazzese, R.; and Davidsson, P. 2017. A role-based approach for orchestrating emergent configuration in the internet of things. In *Proceedings of the 2nd International Workshop on the Internet of Agents (IoA) Workshop*.

Simon, J. 1990. *The Sciences of the artificial*. Cambridge, Mass.: MIT Press.

Valckenars, P.; Brussell, H. V.; and Holvet, T. 2009. Fundamentals of holonic systems and their implications for self-adaptive and self-organizing systems. In *2nd International Conference on Self-Adaptive and Self-Organizing Workshop*, 168–173. IEEE.

Message Validation Pipeline for Agents of the Internet of Everything

Boris Galitsky

Oracle Corp. Redwood Shores CA USA

Oracle Corp. Redwood Shores CA USA
boris.galitsky@oracle.com

Abstract

In the Internet of Everything environment, agents exchange messages, backing up and motivating their decisions. In this environment, validation of message validity, truthfulness, authenticity and consistency is essential. We formulate a problem of domain-independent assessment of argumentation validity based on rhetorical analysis of text. Argumentation structure is discovered in the form of discourse trees extended with edge labels for communicative actions. Extracted argumentation structures are then encoded as defeasible logic programs and are subject to dialectical analysis, to establish the validity of the main claim being communicated. We evaluate the accuracy of each step of this affect processing pipeline as well as overall performance.

Introduction

One of the key features of The Internet of Everything (IoE) is communications in a complex system that includes people, robots and machines. According to (Chambers 2014), IoE connects humans, data, processes and entities to enhance business communication, facilitate employment, well-being, education and healthcare between various communities of people. As billions of people are anticipated to be connected, the requirements of validity and authenticity of textual messages being delivered become essential. To make decisions based, in particular, on textual messages, the claims and their argumentation need to be validated in a domain-independent manner.

Intentional or unintentional untruthful claims and/or their faulty argumentation can lead to an accident, and machines should be able to recognize such claims and their arguments as a part of tackling human errors (Lawless, 2016, Galitsky, 2015). Frequently, human errors are associated with extreme emotions, so we aim at detecting and validating both affective and logical argumentation patterns. Intentional disinformation in a message can also be associated with a security breach (Munro 2017).

When domain knowledge is available and formalized, truthfulness of a claim can be validated directly. However, in most environment it is unavailable and other implicit means need to come into play, such as writing style and writing logic which are domain independent. Hence we attempt to employ the discourse analysis and explore which features of message validation can be leveraged.

When an author attempts to provide a logical or affective argument for something, a number of argumentation patterns can be employed. The basic points of argumentation are reflected in rhetoric structure of text where an argument is presented. A text without argument, with an affective argument and with a logical one would have different rhetoric structures (Moens et al., 2007). When an author uses an affective argument instead of logical arguments, it does not necessarily mean that his argument is invalid. The goal of this study is to explore when an argumentation in an IoT message is valid. We introduce the term of *affective argumentation* to circumscribe a special class of argumentation associated with strong emotions and sentiments.

We select Customer Relationship Management (CRM) as an important domain of IoE. One of the trickiest areas of CRM, involving a number of conflicting agents, is handling customer complaints. In customer complaints, authors are upset with products or services they received, as well as how it was communicated by customer support. Complainants frequently write complaints in a very strong, emotional language, which may distort the logic of argumentation and therefore make a judgment on complaint validity difficult. Both affective and logical argumentation is heavily used.

Especially in banking, customer complaints usually explain what was promised and advertised, and what the customer got. Therefore, a typical complaint arises when a customer attempts to communicate this discrepancy with the bank and does not receive an adequate response. Most complaint authors cite disinformation provided by company agents to avoid accepting responsibility or providing compensation to a customer. At the same time, frequently, customers write complaints attempting to get compensation for allegedly problematic service.

Judging by complaints, most complainants are in genuine distress due to a strong deviation between what they expected from a service, what they received and how it was communicated. Most complaint authors report incompetence, flawed policies, ignorance, indifference to customer needs and misrepresentation from the customer service personnel. The authors have frequently exhausted the communicative means available to them, confused, seeking recommendations from other users and often advise others on avoiding particular financial services. Multiple affective argumentation patterns are used in complaints; the most frequent is an intense description by a complainant on a deviation of what has actually happened from what was expected, according to common sense. This pattern covers both valid and invalid argumentation.

We select the Rhetoric Structure Theory (Rhetoric Structure Theory (RST, Mann and Thompson 1988) as a means to represent discourse features associated with logical and affective argumentation. Nowadays, the performance of both rhetoric parsers and argumentation reasoners has dramatically improved. Taking into account the discourse structure of conflicting dialogs, one can judge on the authenticity and validity of these dialogs in terms of its affective argumentation. In this work we will evaluate the *combined* argument validity assessment system that includes both the *discourse structure extraction* and *reasoning about it* with the purpose of validation of the complainant's claim. Either approach on argument detection from text or on reasoning about formalized arguments has been undertaken, but not the whole text assessment pipeline, required for IoT systems.

Most of the modern techniques treat computational argumentation as specific discourse structures and perform detection of arguments of various sorts in text, such as classifying a text paragraph as argumentative or non-argumentative (Moens et al., 2007). A number of systems recognize components and structures of logical arguments (Sardianos et al., 2015; Stab and Gurevych, 2014). However, these systems do not rely on discourse trees (DTs); they only extract arguments and do not apply logical means to evaluate it. A broad corpus of research deals with logical arguments irrespectively of how they may occur in natural language (Bondarenko et al., 1997). A number of studies addressed argument quality in logic and argumentation theory (van Eemeren et al., 1996; Damer, 2009), however the number of systems that assess the validity of arguments in text is very limited (Cabrio and Villata, 2012; Wei et al., 2016). This is especially true concerning affective argumentation. Most argument mining systems are either classifiers which recognize certain forms of logical arguments in text, or reasoners over logical representation of arguments (Amgoud et al., 2015). Conversely, in this project we intend to build the *whole argumentation pipeline*, augmenting an argument extraction from text with its logical analysis (Fig. 1). This pipeline is necessary to deploy an argumentation analysis in a practical decision support system.

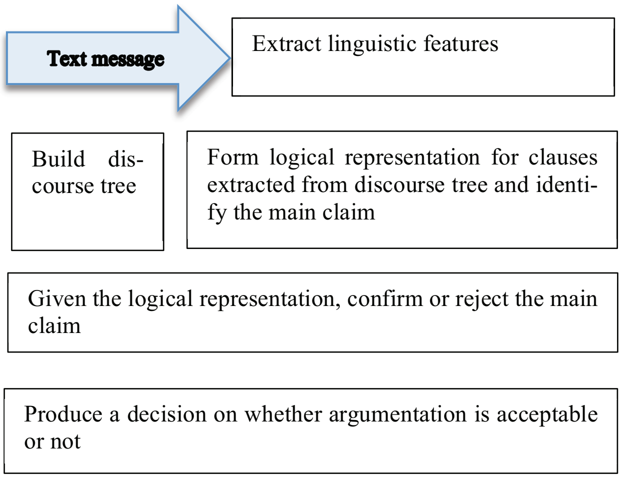

Text message
Extract linguistic features
Build discourse tree
Form logical representation for clauses extracted from discourse tree and identify the main claim
Given the logical representation, confirm or reject the main claim
Produce a decision on whether argumentation is acceptable or not

Figure 1: Claim validity assessment pipeline.

The concept of automatically identifying argumentation schemes was first discussed in (Walton et al., 2008). In (Ghosh et al., 2014) authors investigate argumentation discourse structure of the specific type of communication - online interaction threads. Identifying argumentation in text is connected to the problem of identifying truth, misinformation and disinformation on the web (Pendyala and Figueira, 2015; Galitsky, 2015). In (Lawrence and Reed, 2015) three types of argument structure identification are combined: linguistic features, topic changes and machine learning.

To represent the linguistic features of text, we use the following sources:

1) *Rhetoric relations* between the parts of the sentences, obtained as a *discourse tree*.

2) *Speech acts, communicative actions*, obtained as verbs from the VerbNet resource.

To assess the logical validity of extracted argument, we apply Defeasible Logic Program (DeLP, Garcia and Simari 2004), part of which is built on the fly from facts and clauses extracted from these sources. We integrate argumentation detection and validation components into a decision support system that can be deployed, for example, the CRM domain. To evaluate our approach to extraction and reasoning about argumentation, we choose the dispute resolution / customer complaint validation task because affective argumenation analysis plays an essential role in it.

Representing Argumentative Discourse

We start with a political domain and give an example of conflicting agents providing their interpretation of certain events. These agents provide argumentation for their claims and we will observe how formed rhetoric structures correlate with their argumentation patterns. We focus on Malaysia Airlines Flight 17 example with the agents exchanging affective arguments: *Dutch investigators*, *The Investigative Committee of the Russian Federation,* and *the self-proclaimed Donetsk People's Republic.* It is a controversial conflict where each agent attempts to blame its opponent. Keywords indicating sentiments are underlined. To sound more convincing, each agent does not just formulate its claim, but postulates it in a way to attack the claims of its opponents. To do that, each agent does its best to match the argumentation style of opponents, defeat their claims and apply negative sentiment to them.

"Dutch accident investigators say that strong evidence points to pro-Russian rebels as being fully responsible for shooting down plane. The report indicates where the missile was fired from and identifies who was in control of the territory and pins the downing of MH17 on the pro-Russian rebels." (Fig. 2a)

"The Investigative Committee of the Russian Federation believes that the plane was hit by a missile, which could not be produced in Russia. The committee cites an investigation that established the type of the missile and disagrees with Dutch accident investigators."(Fig. 2)

"Rebels, the self-proclaimed Donetsk People's Republic, deny that they controlled the territory from which the missile was allegedly fired. They confirm that it became possible only after three months after the tragedy to say if rebels controlled one or another town and the claim of Dutch accident investigators is flawed"(Fig. 2c)

To show the structure of arguments one needs to merge discourse relations with speech acts information. We need to know the discourse structure of interactions between agents, and what kind of interactions they are. For argument identification, we don't need to know a domain of interaction (here, aviation), the subjects of these interaction, what are the entities, but we need to take into account mental, domain-independent relations between them. So we need to introduce the concept of Communicative Discourse Tree (CDT).

CDT is a DT with labels for edges that are the VerbNet expressions for verbs (which are communicative actions, CA). Arguments of verbs are substituted from text according to VerbNet frames (Kipper et al., 2008). The first and possibly second argument is instantiated by agents and the consecutive arguments - by noun or verb phrases which are the subjects of CA. For example, the nucleus node for *elaboration* relation (on the left of Fig. 2a) are labeled with *say(Dutch, evidence)*, and the satellite – with *responsible(rebels, shooting_down).* These labels are not intended to express that the subjects of Elementary Discourse Units (EDUs) are *evidence* and *shooting_down* but instead for matching this CDT with others for the purpose of finding similarity between them.

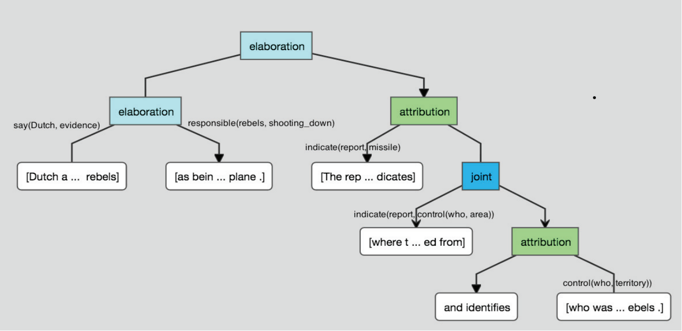


Figure 2a: The claim of the first agent, Dutch accident investigators.

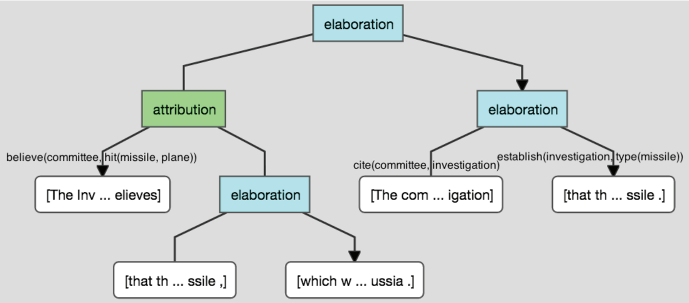


Figure 2b: The claim of the second agent, the Committee.

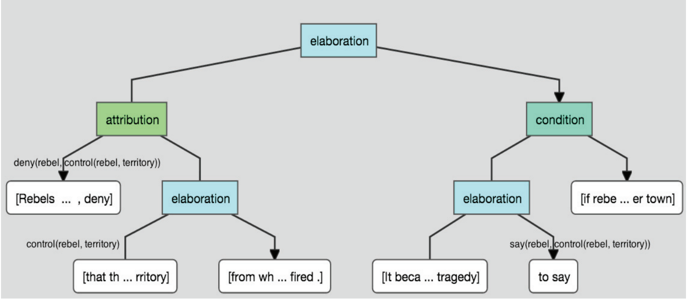


Figure 2c: The claim of the third agent, the rebels.

To summarize, a typical CDT for a text with argumentation includes rhetoric relations other than Elaboration and

Join, and a substantial number of communicative actions. However, these rules are complex enough so that the structure of CDT matters and tree-specific learning is required (Galitsky et al., 2015).

Recognizing Communicative Discourse Trees for Argumentation

Argumentation analysis needs a systematic approach to learn associated discourse structures. The features of CDTs could be represented in a numerical space so that argumentation detection can be conducted; however structural information on DTs would not be leveraged. Also, features of argumentation can potentially be measured in terms of maximal common sub-DTs, but such nearest neighbor learning is computationally intensive and too sensitive to errors in DT construction. Therefore a CDT-kernel learning approach is selected which applies SVM learning to the feature space of all sub-CDTs of the CDT for a given text where an argument is being detected.

Tree Kernel (TK) learning for strings, parse trees and parse thickets is a well-established research area nowadays. The CD-TK counts the number of common sub-trees as the discourse similarity measure between two DTs. A version of TK has been defined for discourse analysis by (Joty and Moschitti, 2014). (Wang et al 2010) used the special form of TK for discourse relation recognition. In this study we extend the TK definition for the CDT, augmenting DT kernel by the information on CAs. TK-based approaches are not very sensitive to errors in parsing (syntactic and rhetoric) because erroneous sub-trees are mostly random and will unlikely be common among different elements of a training set.

A CDT can be represented by a vector V of integer counts of each sub-tree type (without taking into account its ancestors):

V (T) = ($\#of\ subtrees\ of\ type$ 1, … , $\#\ of\ subtrees\ oftype$ I, … , $\#\ of\ subtrees\ of\ type\ n$). This results in a very high dimensionality since the number of different sub-trees is exponential in size. Thus, it is computationally infeasible to directly use the feature vector $\varnothing(T)$. To solve the computational issue, a tree kernel function is introduced to calculate the dot product between the above high dimensional vectors efficiently. Given two tree segments CDT_1 and CDT_2 , the tree kernel function is defined:

$K\ (CDT_1,\ CDT_2) = <V\ (CDT_1\ ),\ V\ (CDT_2\ ) > =\ \Sigma_i\ V\ (CDT_1\ )[i],\ V\ (CDT_2)[i] = \Sigma_{n1}\Sigma_{n2}\ \Sigma_i\ \ I_i(n_1)*\ I_i(n_2),$ where $n_1 \in N_1$, $n_2 \in N_2$ where N_1 and N_2 are the sets of all nodes in CDT_1 and CDT_2 , respectively; I_i (n) is the indicator function: I_i (n) = {1 iff a subtree of type i occurs with root at node; 0 otherwise}. Further details for using TK for paragraph-level and discourse analysis are available in (Galitsky 2017).

Only the arcs of the same type of rhetoric relations (*presentation* relation, such as *antithesis, subject matter* relation, such as *condition, and multinuclear* relation, such as *List*) can be matched when computing common sub-trees. We use *N* for a nucleus or situations presented by this nucleus, and *S* for satellite or situations presented by this satellite. *Situations* are propositions, completed actions or actions in progress, and communicative actions and states (including *beliefs, desires, approve, explain, reconcile* and others). Hence we have the following expression for RST-based generalization '^' for two texts $text_1$ and $text_2$:

$text_1$ ^ $text_2$ = $\cup_{i,j}$ ($rstRelation_{1i,}$ (…,…) ^ $rstRelation_{2j}$ (…,…)), where I ∈ (RST relations in $text_1$), j ∈ (*RST relations in* $text_2$). Further, for a pair of RST relations their generalization looks as follows: $rstRelation_1(N_1,\ S_1)$ ^ $rstRelation_2\ (N_2,\ S_2) = (rstRelation_1$^ $rstRelation_2\ )(\ N_1$^$N_2,\ S_1$^$S_2).$

We define CA as a function of the form *verb (agent, subject, cause),* where *verb* characterizes some type of interaction between involved *agents* (e.g., *explain, confirm, remind, disagree, deny*, etc.), *subject* refers to the information transmitted or object described, and *cause* refers to the motivation or explanation for the subject. To handle meaning of words expressing the subjects of CAs, we apply *word2vec* models (Mikolov et al., 2015).

To compute similarity between the subjects of CAs, we use the following rule. If *subject1=subject2,* then *subject1^subject2 = <subject1, POS(subject1), 1>*. Otherwise, if they have the same part-of-speech, *subject1^subject2=<*,POS(subject1), word2vecDistance(subject1^subject2)>.*

If part-of-speech is different, generalization is an empty tuple. It cannot be further generalized.

We combined Stanford NLP parsing, coreferences, entity extraction, DT construction (discourse parser, Surdeanu et al., 2016 and Joty et al., 2016), VerbNet and Tree Kernel builder into one system available at https://github.com/bgalitsky/relevance-based-on-parse-trees.

Assessing Validity of Extracted Argument Patterns via Dialectical Analysis

To convince an addressee, a message needs to include an argument and its structure needs to be valid. Once an argumentation structure extracted from text is represented via CDT, we need to verify that the main point (target claim) communicated by the author is not logically attacked by her other claims. To assess the validity of the argumentation, a Defeasible Logic Programming (DeLP) approach is selected, an argumentative framework based on logic programming (García and Simari, 2004; Alsinet et al., 2008), and present an overview of the main concepts associated with it.

A DeLP is a set of facts, strict rules Π of the form (A:-B) , and a set of defeasible rules Δ of the form A-<B, whose intended meaning is "if B is the case, then usually A is also the case". Let P=(Π, Δ) be a DeLP program and L a ground literal.

Let us now build an example of a DeLP for legal reasoning about facts extracted from text (Fig. 3a). A judge hears an eviction case and wants to make a judgment on whether rent was provably paid (deposited) or not (denoted as *rent_receipt).* An input is a text where a defendant is expressing his point. Underlined words form the clause in DeLP, and the other expressions formed the facts (Fig. 3b).

The landlord contacted me, the tenant, and the rent was requested. However, I refused the rent since I demanded repair to be done. I reminded the landlord about necessary repairs, but the landlord issued the three-day notice confirming that the rent was overdue. Regretfully, the property still stayed unrepaired

Defeasible Rules Prepared In Advance
rent_receipt -< rent_deposit_transaction.
rent_deposit_transaction -< contact_tenant.
┐ *rent_deposit_transaction -<contact_tenant,*
three_days_notice_is_issued.
┐ *rent_deposit_transaction -< rent_is_overdue.*
┐ *repair_is_done -< rent_refused, repair_is_done.*
repair_is_done -< rent_is_requested.
┐ *rent_deposit_transaction -<*
tenant_short_on_money, repair_is_done.
┐ *repair_is_done -< repair_is_requested.*
┐ *repair_is_done -<rent_is_requested.*
┐ *repair_is_requested -< stay_unrepaired.*
┐ *repair_is_done -< stay_unrepaired.*
Target Claim to be Assessed
? - rent_receipt
Clauses Extracted from text
repair_is_done -< rent_refused.
Facts from text
contact_tenant. rent_is_requested. rent_refused. remind_about_repair. three_days_notice_is_issued. rent_ is_overdue. stay_unrepaired.

Figure 3a: An example of a Defeasible Logic Program for modeling category mapping.

A *defeasible derivation* of L from P consists of a finite sequence $L_1, L_2, \ldots, L_n = L$ of ground literals, such that each literal L_i is in the sequence because:
(a) L_i is a fact in Π, or
(b) there exists a rule R_i in P (strict or defeasible) with head L_i and body $B_1, B_2, \ldots, B_k$ and every literal of the body is an element L_j of the sequence appearing before L_j ($j < i$).

Let h be a literal, and P=(Π, Δ) a DeLP program. We say that <A, h> is an *argument* for h, if A is a set of defeasible rules of Δ, such that:
1. there exists a defeasible derivation for h from (Π ∪ A);
2. the set (Π ∪ A) is non-contradictory; and
3. A is minimal: there is no proper subset A_0 of A such that A_0 satisfies conditions (1) and (2).
Hence an argument <A, h> is a minimal non-contradictory set of defeasible rules, obtained from a defeasible derivation for a given literal h associated with a program P.

We say that $<A_1, h_1>$ *attacks* $<A_2, h_2>$ iff there exists a sub-argument <A, h> of $<A_2, h_2>$ $(A \subseteq A_1)$ such that h and h_1 are inconsistent (i.e. $\Pi \cup \{h, h_1\}$ derives complementary literals). We will say that $<A_1, h_1>$ *defeats* $<A_2, h_2>$ if $<A_1, h_1>$ attacks $<A_2, h_2>$ at a sub-argument <A, h> and $<A_1, h_1>$ is strictly preferred (or not comparable to) <A, h>. In the first case we will refer to $<A_1, h_1>$ as a *proper defeater*, whereas in the second case it will be a *blocking defeater*. Defeaters are arguments which can be in their turn attacked by other arguments, as is the case in a human dialogue. An *argumentation line* is a sequence of arguments where each element in a sequence defeats its predecessor. In the case of DeLP, there are a number of *acceptability* requirements for argumentation lines in order to avoid fallacies (such as circular reasoning by repeating the same argument twice).

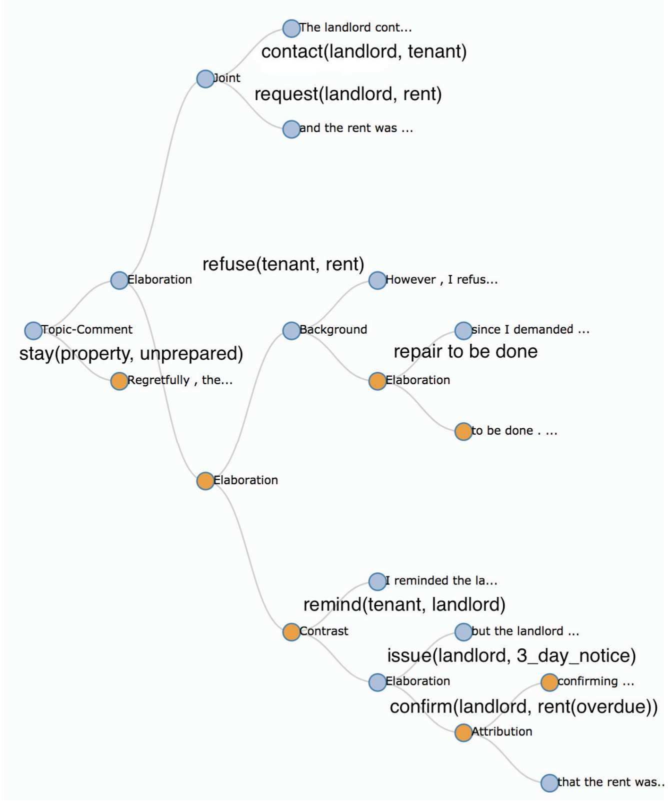


Figure 3b: Text of a complaint and its CDT (visualization by Joty et al., 2013).

Target claims can be considered DeLP queries which are solved in terms of dialectical trees, which subsumes all possible argumentation lines for a given query. The definition of dialectical tree provides us with an algorithmic view for discovering implicit self-attack relations in users' claims. Let <A_0, h_0> be an argument (target claim) from a program P. A *dialectical tree* for <A_0, h_0> is defined as follows:

1. The root of the tree is labeled with <A_0, h_0>
2. Let N be a non-root vertex of the tree labeled <A_n, h_n> and Λ=[<A_0, h_0>, <A_1, h_1>, …, <A_n, h_n>] (the sequence of labels of the path from the root to N). Let [<B_0, q_0>, <B_1, q_1>, …, <B_k, q_k>] all attack <A_n, h_n>.

For each attacker <B_i, q_i> with acceptable argumentation line [Λ,<B_i, q_i>], we have an arc between N and its *child* N_i.

A labeling on the dialectical tree can be then performed as follows:

1. All leaves are to be labeled as U-nodes (undefeated nodes).
2. Any inner node is to be labeled as a U-node whenever all of its associated children nodes are labeled as D-nodes.
3. Any inner node is to be labeled as a D-node whenever at least one of its associated children nodes is labeled as U-node.

After performing this labeling, if the root node of the tree is labeled as a U-node, the original argument at issue (and its conclusion) can be assumed as *justified* or *warranted.*

In our DeLP example, the literal *rent_receipt* is supported by *<A, rent_receipt>* = <{ (*rent_receipt -< rent_deposit_transaction*), (*rent_deposit_transaction -< tenant_short_on_money*)}, *rent_receipt*> and there exist three defeaters for it with three respective argumentation lines: *<B$_1$, ┐rent_deposit_transaction>* = <{(┐*rent_deposit_transaction -<*
tenant_short_on_money,
three_days_notice_is_issued)},
rent_deposit_transaction>.

<B$_2$,┐ *rent_deposit_transaction>* =
<{(┐ *rent_deposit_transaction -<*
tenant_short_on_money, repair_is_done),
(*repair_is_done -< rent_refused*) },
rent_deposit_transaction>.

<B$_3$, ┐*rent_deposit_transaction>* = <{(┐*rent_deposit_transaction -< rent_is_overdue*)}, *rent_deposit_transaction>.* The first two are proper defeaters and the last one is a blocking defeater. Observe that the first argument structure has the counter-argument,
<{*rent_deposit_transaction -<*
tenant_short_on_money},
rent_deposit_transaction), but it is not a defeater because the former is more specific. Thus, no defeaters exist and the argumentation line ends there.

B$_3$ above has a blocking defeater <{(*rent_deposit_transaction -<*
tenant_short_on_money)},
rent_deposit_transaction> which is a disagreement sub-argument of *<A, rent_receipt>* and it cannot be introduced since it gives rise to an unacceptable argumentation line. *B$_2$* has two defeaters which can be introduced: *<C$_1$,* ┐*repair_is_done >, where C$_1$* = {(┐*repair_is_done -< rent_refused,*
repair_is_done),
(*repair_is_done -< rent_is_requsted*)}, a proper defeater, and *<C$_2$,* ┐*repair_is_done >,* where *C$_2$*={(┐*repair_is_done -< repair_is_requested*)} is a blocking defeater. Hence one of these lines is further split into two; C$_1$ has a blocking defeater that can be introduced in the line
<D$_1$, ┐*repair_is_done >,* where *D$_1$*= <{(┐*repair_is_done -< stay_unrepaired*)}. *D$_1$* and *C$_2$* have a blocking defeater, but they cannot be introduced because they make the argumentation line inacceptable. Hence the state *rent_receipt* cannot be reached, as the argument supporting the literal *rent_receipt,* is not warranted. The dialectical tree for *A* is shown in Fig. 4.

Having shown how to build a dialectic tree, we are now ready to outline the algorithm for validation of the domain-specific claim for arguments extracted from text:

1. Build a DT from input text;
2. Attach communicative actions to its edges to form CDT;
3. Extract subjects of communicative actions attached to CDT and add to 'Facts' section;
4. Extract the arguments for rhetoric relation *contrast* and communicative actions of the class *disagree* and add to 'Clauses Extracted FromText' section;
5. Add a domain-specific section to DeLP;
6. Having the DeLP formed, build a dialectical tree and assess the claim.

We used (Tweety 2017) system for DeLP implementation.

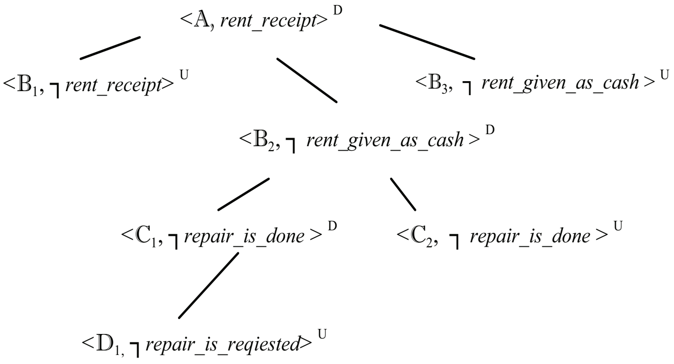


Figure 4: Dialectical tree for target claim rent_receipt.

Intense Arguments Dataset

The purpose of this dataset is to collect texts where authors do their best to bring their points across by employing all means to show that they are right and their opponents are wrong. Complainants are emotionally charged writers who describe problems they encountered with a financial service and how they attempted to solve it.

Most complaint authors report incompetence, flawed policies, ignorance, indifference to customer needs and misrepresentation from the customer service personnel (Galitsky et al., 2009). The focus of a complaint is a proof that the proponent is right and her opponent is wrong, followed by a resolution proposal and a desired outcome.

Complaints reveal shady practices of banks during the financial crisis of 2007, such as manipulating an order of transactions to charge a highest possible amount of non-sufficient fund fees. Moreover, banks attempted to communicate this practice as a necessity to process a wide amount of checks. This is the most frequent topic of customer complaints, so one can track a manifold of argumentation patterns applied to this topic.

For a given topic such as *insufficient funds fee*, this dataset provides many distinct ways of argumentation that this fee is unfair. Therefore, our dataset allows for systematic exploration of the topic-independent clusters of argumentation patterns and observe a link between argumentation type and overall complaint validity. Other argumentation datasets including legal arguments, student essays, Internet argument corpus, fact-feeling, and political debates have a strong variation of topics so that it is harder to track a spectrum of possible argumentation patterns per topic. Unlike professional writing in legal and political domains, the messages produced by complainants have a simple motivational structure, a transparency of their purpose and occurs in a fixed domain and context. In our dataset, the affective arguments play a critical rule for the well-being of the authors, subject to an unfair charge of a large amount of money or eviction from home. Therefore, the authors attempt to provide as strong argumentation as possible to back up their claims and strengthen their case.

Evaluation of Detection and Validation of Affective Arguments

The objective of argument detection task is to identify all kinds of arguments, not only ones associated with customer complaints. We formed the *positive* dataset from textual customer complaints dataset (Galitsky et al., 2009, and https://github.com/bgalitsky/relevance-based-on-parse-trees/blob/ master/src/test/resources/opinionsFinanceTagged.xls.zip. scraped from consumer advocacy site PlanetFeedback.com. This dataset is used for both argument detection and argument validity tasks .

Table 1: Evaluation results for argument detection.

Method / sources	P	R	F1
Bag-of-words	57.2	53.1	55.07
WEKA-Naïve Bayes	59.4	55.0	57.12
SVM TK for RST and CA (full parse trees)	77.2	74.4	75.77
SVM TK for DT	63.6	62.8	63.20
SVM TK for CDT	82.4	77.0	79.61

For the *negative* dataset, only for the affective argument detection task, we used Wikipedia, factual news sources, and also the component of (Lee, 2001) dataset that includes such sections of the corpus as: [‘tells’], instructions for how to use software; [‘tele’], instructions for how to use hardware, and [news], a presentation of a news article in an objective, independent manner, and others. Further details on the data set are available in (Galitsky et al 2015).

A baseline approach relies on keywords and syntactic features to detect argumentation (Table 1). Frequently, a coordinated pair of communicative actions (so that at least one has a negative sentiment polarity related to an opponent) is a hint that logical argumentation is present. This naïve approach is outperformed by the top performing TK learning CDT approach by 29%. SVM TK of CDT outperforms SVM TK for RST+CA and RST + full parse trees (Galitsky, 2017) by about 5% due to noisy syntactic data which is frequently redundant for argumentation detection.

SVM TK approach provides acceptable F-measure but does not help to explain how exactly the affective argument identification problem is solved, providing only final scoring and class labels. Nearest neighbor maximal common subgraph algorithm is much more fruitful in this respect (Galitsky et al., 2015). Comparing the bottom two rows, we observe that it is possible, but infrequent to express an affective argument without CAs.

Assessing logical arguments extracted from text, we were interested in cases where an author provides invalid, inconsistent, self-contradicting cases. That is important for CRM systems focused on customer retention and facilitating communication with a customer (Galitsky et al 2009). The domain of residential real estate complaints was selected and a DeLP thesaurus was built for this domain. Automated complaint processing system can be essential, for example, for property management companies in their decision support procedures (Constantinos et al., 2003).

Table 2: Evaluation results for argument validation.

Types of complaints	P	R	F1 of validation	F1 of total
Single rhetoric relation of type *contrast*	87.3	15.6	26.5	18.7
Single communicative action of type *disagree*	85.2	18.4	30.3	24.8
Two or three specific relations or communicative actions	80.2	20.6	32.8	25.4
Four and above specific relations or communicative actions	86.3	16.5	27.7	21.7

In our validity assessment we focus on target features related to how a given complaint needs to be handled, such as *compensation_required, proceed_with_eviction, rent_receipt* and others.

Validity assessment results are shown in Table 2. In the first and second rows, we show the results of the simplest complaint with a single rhetoric relation such as *contrast* and a single CA indicating an extracted argumentation attack relation respectively. In the third row we assess complaints of average complexity, and in the bottom row, the most complex, longer complaints in terms of their CDTs. The third column shows detection accuracy for invalid argumentation in complaints in a stand-alone argument validation system. Finally, the fourth column shows the accuracy of the integrated argumentation extraction and validation system.

Recall is low because in the majority of cases the invalidity of claims is due to factors other than being self-defeated. Precision is relatively high since if a logical flaw in an argument is established, most likely the whole claim is invalid because other factors besides argumentation (such as false facts) contribute as well. As complexity of a complaint and its discourse tree grows, F1 first improves since more logical terms are available and then goes back down as there is a higher chance of a reasoning error due to a noisier input.

For decision support systems, it is important to maintain a low false positive rate. It is acceptable to miss invalid complaints, but for a detected invalid complain, confidence should be rather high. If a human agent is recommended to look at a given complaint as invalid, her expectations should be met most of the time. Although F1-measure of the overall argument detection and validation system is low in comparison with modern recognition systems, it is still believed to be usable as a component of a CRM decision support system.

Conclusions

In this study we explored a possibility to validate messages in an IoE environment. We observed that by relying on discourse tree data, one can reliably detect patterns of logical and affective argumentation. Communicative discourse trees become a source of information to form a defeasible logic program to validate an argumentation structure. Although the performance of the former being about 80% is significantly above that of the latter (29%), the overall pipeline can be useful for detecting cases of invalid affective argumentation, which are important in decision support for CRM.

To the best of our knowledge, this is the first study building the whole argument validity pipeline, from text to a validated claim in it, which is a basis of IoE decision support. Hence although the overall argument validation accuracy is fairly low, there is no existing system to compare this performance against.

In this paper, to support IoE message validation, we attempted to combine the best of both worlds, argumentation mining from text and reasoning about the extracted argument. Whereas applications of either technology are limited, the whole argumentation pipeline is expected to find a broad range of applications. In this work we focused on a very specific legal area such as customer complaints, but it is easy to see a decision support system employing the proposed argumentation pipeline in other domains of CRM.

An important finding of this study is that argumentation structure can be discovered via the features of extended discourse representation, combining information on how an author organizes her thoughts with information on how involved agents communicate these thoughts. Once a communicative discourse tree is formed and identified as being correlated to argumentation, a defeasible logic program can be built from this tree and the dialectical analysis can validate the main claim.

Although validating agents' messages, affective argument should not be confused with an *appeal to emotion*, a logical fallacy characterized by the manipulation of the recipient's emotions in order to win an argument, especially in the absence of factual evidence. This kind of appeal to emotion is a type of red herring and encompasses several logical fallacies.

References

Amgoud, L., Besnard, P. and Hunter, A. 2015. Representing and Reasoning About Arguments Mined from Texts and Dialogues. ECSQARU, pp 60-71.

Alsinet, T., Carlos Iván Chesñevar, Lluis Godo, Guillermo Ricardo Simari. 2008. A logic programming framework for possibilistic argumentation: Formalization and logical properties. Fuzzy Sets and Systems 159(10): 1208-1228.

Bondarenko, A., Dung, P., Kowalski, R., Toni, F. 1997. An abstract, argumentation-theoretic approach to default reasoning. Artificial Intelligence 93, pp 63–101.

Cabrio, E. and Villata, S. 2012. Combining textual entailment and argumentation theory for supporting online debates interactions. In ACL.

Chali, Y. Shafiq R. Joty, and Sadid A. Hasan. 2009. Complex question answering: unsupervised learning approaches and experiments. J. Artif. Int. Res. 35, 1 (May 2009), 1-47.

Chambers, J. (2014, 1/15), Are you ready for the Internet of everything? World Economic Forum, from https://www.weforum.org/agenda/2014/01/are-you-ready-for-the-internet-of-everything/

Constantinos JS, Sarmaniotis, C., Stafyla, A. CRM and customer-centric knowledge management: an empirical research. 2003. Business Process Management Journal, Vol. 9, Issue: 5, 617-634.

Damer, T.E. 2009. Attacking Faulty Reasoning: A Practical Guide to Fallacy-Free Reasoning. Wadsworth Cengage Learning.

Feng, WV and Hirst, G. 2014. A linear- time bottom-up discourse parser with constraints and post-editing. In *ACL.*

Galitsky, B., MP González, CI Chesñevar. 2009. A novel approach for classifying customer complaints through graphs similarities in argumentative dialogue. Decision Support Systems, 46-3, 717-729.

Galitsky, B. 2012. Machine learning of syntactic parse trees for search and classification of text. *Engineering Application of AI* , 26(3) 1072-91.

Galitsky, B, Ilvovsky, D. and Kuznetsov SO. 2015. Rhetoric Map of an Answer to Compound Queries Knowledge Trail Inc. ACL-2, 681–686.

Galitsky, B., Detecting Rumor and Disinformation by Web Mining. 2015. AAAI Spring Symposium.

Galitsky, B. 2017. Using Extended Tree Kernel to Recognize Metalanguage in Text. In Uncertainty Modeling, Volume 683 of the series Studies in Computational Intelligence, pp.71-96, Springer.

Garcia, A. and Simari GR. 2004. Defeasible Logic Programming: An Argumentative Approach. Theory and Practice of Logic Programming 4(1-2):95--138.

Ghosh, Debanjan, Smaranda Muresan, Nina Wacholder, Mark Aakhus, and Matthew Mitsui. 2014. Analyzing argumentative discourse units in online interactions. In Proceedings of the First Workshop on Argumentation Mining, pages 39–48, Baltimore, Maryland, June. ACL.

Joty, Shafiq R, Giuseppe Carenini, Raymond T Ng, and Yashar Mehdad. 2013. Combining intra-and multi- sentential rhetorical parsing for document-level dis- course analysis. In *ACL (1)*, pages 486–496.

Joty, Shafiq R and A. Moschitti. 2014. Discriminative Reranking of Discourse Parses Using Tree Kernels. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), page

Kipper, K. Korhonen, A., Ryant, N. and Palmer, M. 2008. A large-scale classification of English verbs. Language Resources and Evaluation Journal, 42, pp. 21-40.

Lawrence, J. and C. Reed Combining Argument Mining Techniques. ArgMining@ HLT-NAACL, 127-136.

Mann, William and Sandra Thompson. 1988. Rhetorical structure theory: Towards a functional theory of text organization. Text-Interdisciplinary Journal for the Study of Discourse, 8(3):243–281.

Mikolov, Tomas, Chen, Kai, Corrado; G.S., Dean; Jeffrey (2015). Computing numeric representations of words in a high-dimensional space. US Patent 9,037,464, Google, Inc.

Moens, Marie-Francine, Erik Boiy, Raquel Mochales Palau, and Chris Reed. 2007. Automatic detection of arguments in legal texts. In Proceedings of the 11th International Conference on Artificial Intelligence and Law, ICAIL '07, pages 225–230, Stanford, CA, USA.

Munro, K. 2017. How to beat security threats to 'internet of things', from http://www.bbc.com/news/av/technology-39926126/how-to-beat-security-threats-to-internet-of-things.

Pendyala, V.S., Figueira, S. 2015. Towards a truthful world wide web from a humanitarian perspective. Global Humanitarian Technology Conference,2015 8-11.

Sardianos, C. Katakis, IM, Petasis, G. and Karkaletsis, V. 2015. Argument extraction from news. In Proceedings of the 2nd Workshop on Argumentation Mining, pages 56–66, Denver, CO, USA.

Stab, C. and Gurevych, I. 2014. Identifying argumentative discourse structures in persuasive essays. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP '14, pages 46–56, Doha, Qatar.

Surdeanu, Mihai, Thomas Hicks, and Marco A. Valenzuela-Escarcega. Two Practical Rhetorical Structure Theory Parsers. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies: Software Demonstrations (NAACL HLT), 201

Tweety 2016. Last downloaded Dec 12, 2016.

https://javalibs.com/artifact/net.sf.tweety.arg/delp.

Thimm, M. 2014. Tweety - A Comprehensive Collection of Java Libraries for Logical Aspects of Artificial Intelligence and Knowledge Representation. In Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR'14). Vienna.

van Eemeren, Frans H., Rob Grootendorst, and Francisca Snoeck Henkemans. 1996. Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments. Routledge, Taylor & Francis Group.

Walton, D. N., Reed, C., & Macagno, F. (2008) Argumentation Schemes. Cambridge: Cambridge University Press.

Wang, W., Su, J., Tan, C.L. 2010. Kernel Based Discourse Relation Recognition with Temporal Ordering Information. In *ACL.*

Policy Issues Regarding Implementations of Cyber Attack Resilience Solutions for Cyber Physical Systems

Barry M. Horowitz
Munster Professor of Systems and Information Engineering
University of Virginia, Charlottesville, Virginia 22904
bh8e@virginia.edu

Abstract

The Internet of Things (IoT) is dramatically increasing complexity in cities, commerce and homes. This complexity is increasing the risk to cyber threats. To reduce these risks, resilient cyberphysical systems must be able to respond to different types of disturbances (errors; cyberattacks). Organizational, system and infrastructure security pose new challenges for policy considerations that reduce cyber risks rather than simply reacting to cyberattacks. Indeed, policies must be crafted to require anticipatory responses able to discriminate between anomalies caused by errors and those driven by cyberattackers for malicious purposes that may result in obvious damage (e.g., equipment destruction, injury or death) or subtle control (e.g., Stuxnet). We conclude that anticipatory resilience solutions for cyberphysical systems will require teams of government and commercial organizations to address the consequences of cyberattacks, to detect them and to defend against them.

Introduction: Context

A resilient cyber physical system is one that maintains state awareness and an accepted level of operational normalcy in response to disturbances, including threats of an unexpected and malicious nature (Rieger et al., 2009). Responding to cyber attacks against cyber physical systems such as automated vehicles, weapon systems, and manufacturing systems requires addressing cyber attack risks that can potentially include consequences such as injuries or death. The difference in the severity of these consequences compared to those of information system cyber attacks brings with it new policy considerations related to cybersecurity. However, as was the case for the integration of information systems through the Internet, unless special attention is paid to this matter early on, security will likely be dominated by responses to actual attacks, rather than anticipatory solutions designed to reduce the risks.

Over the past seven years, the author has been leading a technology-focused research effort that addresses cyber attack resilience for physical systems (Jones et al., 2012; Jones et al., 2013; Horowitz & Pence, 2013; Bayuk & Horowitz, 2011; Gay et al., 2017; Babineau et al., 2012; Jones et al., 2011; Horowitz, 2016; Horowitz & Lucero, 2017). Unlike cyber attack defense solutions, resilience solutions involve monitoring to detect successful cyber attacks and support for rapid reconfiguration of the attacked system for continued operation with contained consequences. The reconfigurations can include modifications in the roles and procedures for human system operators as well as technology related adjustments. The monitoring sub-system(s), referred to as a Sentinel, for detection of attacks and derivation of potential reconfigurations must be very highly secured to avoid becoming an attractive target for attacks. Note that resilience solutions can serve as a deterrent to attackers since they promise to reduce the highest risk consequences of potential cyber attacks. As an example of cyber attack resiliency, consider an automobile equipped with an automated collision avoidance capability. A variety of cyber attacks have been demonstrated in which an automobile could be automatically directed toward a possible collision with another nearby vehicle. Monitoring the automobile's sensor outputs, control system inputs and outputs, and driver inputs through the acceleration and brake pedals, would provide a basis for recognition of an inconsistency potentially caused by a cyber-attack impacting the control system. However, the control error could also be the result of erroneous sensor inputs. Comparing measurements from a diverse set of sensors would provide a basis for detecting and responding to either a failed or cyber attacked sensor sub-system. Integration of the alternate explanations for the control error provides the opportunity to automatically correct the situation or alternately, provide opportunity for the driver to respond. Note that a resilience solution impacts the effectiveness of a variety of possible cyber attacks that would create common symptoms.

The technology-focused research effort has included a number of prototyping projects involving protection of currently available, highly automated physical systems that are being cyber-attacked. These prototyping activities have served to demonstrate the importance of, and potential for, cyber attack resiliency solutions. Specific operational prototyping activities have included: 1) a DoD-sponsored effort involving cyber defense of an unmanned air vehicle (UAV) conducting surveillance missions (including inflight evaluations) Miller, 2014a, 2) defending automobiles (including Virginia State Police exercises with unsuspecting policemen driving cyber attacked police cars) (NBC.29, 2015; Higgins, 2015a), and 3) a National Institute of Standards sponsored effort involving the defense of a 3D Printer through the monitoring of its motors, temperature controllers and other physical component controllers, while in the process of printing defective parts due to cyber-attacks on the machine's internal technology components. These real-world cases have served to illuminate a number of important and complex policy issues made visible to government and industry participants involved with the prototype projects. These policy issues are the subjects of this paper.

The Need to Address Cybersecurity for Physical Systems

Two important, closely related technology trends are occurring simultaneously; however, the two trends are not reinforcing.

Trend 1: The integration of technology-based automation capabilities associated with physical systems. This trend includes:

- Development of autonomous and highly automated vehicles for transportation (air, ground and sea)
- Development of increasingly-capable 3D printers and robots for manufacturing
- Use of network-based access to physical systems to enable remote control and/or monitoring (e.g., physical system maintenance plans based upon measured conditions of use, customized patient health care related responses based upon collected information from on-body sensors)
- Emergent Internet of Things (IoT) opportunities that relate to consumer products, the home, smart cities, etc.

Trend 2: The increasing recognition of the potential risks related to cyber attacks on physical systems, particularly with regard to human safety, not typically associated with cyber attacks on conventional information systems. While attacks on physical systems have not yet emerged as a high risk, various technology demonstrations have shown the potential threat of these types of attacks. Such demonstrations include the following:

- Recent automobile attacks (Higgins, 2015b) showing the feasibility of cyber attacks to cause physical harm.
- Actual high visibility cyber attacks on physical systems, such as the Stuxnet attacks (Falliere et al., 2011) highlighting the potential for other attacks of this kind. The Stuxnet attacks impacted a large number of Iranian nuclear reactors, serving as a warning that industrial computer-controlled physical systems are vulnerable to attack.
- Less publicized attacks on physical systems that have also occurred. For example, a German government security report indicated that an unnamed steel plant suffered an attack that impacted its blast furnace, causing significant damage (CART, 2013).

To-date, the cybersecurity engineering community has principally been focused on information systems, an area where the risks are different and the technical factors regarding cyber defense pose significantly different challenges.

Historic Patterns for Addressing Cybersecurity

While cybersecurity experts point to the fact that anticipatory design of cybersecurity features into systems provides a pathway for achieving better security, historically most solutions have been add-ons to systems in response to actual attacks (Miller, 2014b). The reasons for this are economic. When new innovations are in their early development phase (such as autonomous vehicles), designers are consumed with achieving a working system, and security is treated as something that will follow. When the innovation is ready to bring to market, concern about the cost impacts of security on the new products' prices further delays security implementation. When the new products are selling, but significant attacks have yet to occur, there is no pressing demand to anticipate attacks. When attacks start occurring, and there are already large numbers of existing systems in use, responsive patching becomes the de facto solution.

For existing information systems, the major consequences of cyber attacks have been financial in nature or related to privacy. Should human safety become a primary risk of cyber-attacks in the future, new societal patterns may emerge that demand stronger anticipatory solutions. Anticipatory solutions must be designed not only on the basis of prior attacks, but also based upon predictions of what cyber attackers might target in the future and how they might implement these attacks. Prediction of attacker behavior is quite complex, requiring considerations such as: 1) historic attacks, 2) attacker motivations; 3) attack complexity and corresponding attacker skill requirements; 4) costs of design and implementation; 5) risks of attacks failing; and 6) risks of getting caught. This situation is exacerbated by the need for competitors to share information (e.g., historic attack information) in order to have a more complete basis for making predictions and to provide the opportunity to derive a common framework for considering solutions that are related to a domain of similar products. Furthermore, for physical systems classes that include rapidly changing automation features, predictions can be unstable (e.g., the increasing rate for adding new automation features in automobiles points to the need for annual reconsideration of potential cyber attacks and the corresponding defenses). This situation is further complicated by the fact that it would be difficult to measure the success of resilience solutions serving to deter attacks, since deterrence is not directly observable. For all of these reasons, one can expect that managing the design of anticipatory defenses would be quite difficult. Furthermore, should successful, high-visibility cyber-attacks occur, confidence in anticipatory solutions serving as a deterrent would likely suffer, thereby resulting in reconsiderations regarding their effectiveness.

In the event that more emphasis is placed on implementing anticipatory solutions to cyber-attacks, questions arise regarding the roles of industry and government in deciding on specific resilience requirements. With its superior knowledge of physical system design details and potential means of exploiting those details, industry is in a much stronger position than government to address the selection of anticipatory solutions. On the other hand, with its access to information regarding actual cyber-attacks, along with our country's history of relying on government for implementing safety measures, government does possess some advantages. This suggests a shared role, but a variety of cybersecurity-specific complications, discussed below, emerge when dividing accountabilities.

To demonstrate policy issues regarding the anticipation of cyber-attacks, we return to the automobile collision avoidance system scenario described in the initial section of this article. Note that this automobile example is pertinent to other classes of physical systems. Assume that a collision event were to actually occur as a result of the earlier-described cyber attack. Members of the law enforcement community would be the principal investigators as to cause, but they would have no basis for determining the cause as being a cyber attack. Doing so would likely require access to a portion of the stored data from the involved automobiles' onboard systems. Depending on the specific manufacturers and models of the involved automobiles, the data required to identify the cause as a cyber attack would likely vary from vehicle to vehicle. Due to these variations, the costs associated with necessary field tools and officer training would be driven up. This may suggest standardization as a needed solution, but the standardization of pertinent data implies corresponding commonalities in the designs of automation features, which creates issues related to competition. To further complicate matters, the cybersecurity community recognizes risks associated with "monoculture solutions"; i.e., common designs are vulnerable to common cyber attacks, enabling undesirable reuse opportunities by those who employ or sell software that accomplishes cyber attacks. In addition, the automobile companies and individual drivers may be reticent to provide such data (e.g., Intellectual Property protection reasons, and privacy reasons unrelated to the incident). This very complex set of circumstances will require significant attention and government and industry collaboration. Yet without evidence that cyber attacks on automobiles are actually occurring, it would take very strong leadership to push through measures allowing law enforcement to address cyber attacks on automobiles in an anticipatory manner.

Recognizing the natural desire to avoid costs associated with anticipating cybersecurity, perhaps historical roles in safety regulation can provide a starting point for government involvement. Historically, with certain exceptions, safety analyses have not considered cyber attacks as a safety issue. The trend of advancing highly automated physical systems into general use raises the issue of whether or not the safety communities (government and industry) should start to address this intersection. In doing so, it becomes necessary to understand and account for the relationships between the systems at risk and other interconnected and interrelated systems that can be a pathway for generating a cyber attack. If one starts down this path, some new and complex issues arise.

Mission-Based Cybersecurity

In this section, an integrated set of interconnected systems' combined mission is considered as the point of departure regarding anticipation of cyber attacks. The technology-focused research efforts that the author has been engaged with have addressed a number of illuminating scenarios. For example, as part of addressing UAV cybersecurity solutions, a variety of potential cyber attacks were considered as potential concerns that call for defensive capabilities. For illustration purposes, consider cyber attacks aimed at modifying a UAV's flight path, adversely impacting its ability to carry out its safety-related surveillance mission (e.g., monitoring an oil or gas pipeline). Such an attack could, for example, accompany a physical attack on the pipeline. One way for an attacker to accomplish this outcome is to modify mission-related waypoints that have been entered into the navigation system on board the aircraft. One possible solution addresses a cyber-attack in which the ground-based portion of the UAV system is utilized by the attacker to automatically send surveillance-disrupting changes to the navigation waypoints loaded on board the aircraft. These changes would cause the aircraft to be routed in a manner that prevents gathering of the critical information the mission was intended to collect. A potential solution could involve monitoring the aircraft's navigation system and the pilot's data entry system (e.g., key stroke monitoring). If, when a change in waypoint is detected on the aircraft, there is no corresponding pilot data input, then a cyber attack is a possible cause. In response, the aircraft could transmit information to designated personnel who could then take actions to confirm and address the cyber attack possibility. This example highlights the fact that certain attack detections require coordinating information retrieved from multiple subsystems at different locations. If one considers air traffic control systems, a parallel set of circumstances can occur involving ground-based subsystems (e.g., surveillance, communications, navigation, air traffic controller support systems) and corresponding airborne subsystems. Implementation of solutions would require decisions regarding the perceived level of risk, solution costs, the allocation of costs to subsystems, and decisions regarding the sources for paying for the solutions. Furthermore, for certain attacks that can create the same outcomes through different points of insertion, our technology-focused research efforts have shown that the ease of attack on one subsystem can be very different from that of another subsystem, providing opportunities to address the minimization of total costs when dealing with high priority targets. However, lowering total costs can bring with it controversial cost allocation issues, requiring policies that manage such situations. As stated earlier, without prior data that provides evidence that relevant cyber attacks are actually occurring, it will very take strong leadership to address the issues of anticipating safety-related outcomes and cost allocation for implementation of solutions.

Education of Engineers and Policy-Makers

The discussions presented above do not address what may be the most critical issue in implementing cybersecurity for physical systems, namely the education of both our engineering and policy-making communities. Teams that include mechanical, electrical, and system engineers design physical systems. Engineering schools do not integrate computer security courses into the individual curriculums of these engineering disciplines. As a result, there are a very limited number of physical system design engineers who have the requisite knowledge to design systems that better account for cybersecurity considerations. Furthermore, educators in these areas of engineering have no historic basis for engaging in the cybersecurity-related aspects of their fields. As a result, our colleges and universities need to consider this emergent need and develop cross-department programs that are responsive to this new, important requirement. Development of new programs can be influenced by a strong calling from industry to the education system, including providing financial support for development of new integrated programs, student internships, and professional education programs that support their current workforce. Similar to the issues discussed earlier, it will take strong industry leadership to support such programs without prior data providing evidence that cyber-attacks on physical systems are occurring.

A similar situation faces the policy-making community. As part of structuring resilience-related prototyping efforts, researchers have to address project-specific safety issues associated with conducting experiments. This requirement calls for interactions with a variety of policy organizations. Based on such interactions, it became clear to the author that the imagination of policy-makers with regards to what cyber-attacks could potentially accomplish far exceeded reality. Furthermore, discussions surrounding particular cyber-attacks and their consequences, as well as the solutions to be evaluated, made clear that the requisite technology-related knowledge became an issue in deriving safety controls. Interestingly, in some cases, the policy outcomes could have been unnecessarily conservative and in others, not conservative enough. Another important finding was that that the policy community found that the security community was greatly steeped in specialized technical jargon, providing a barrier to beneficial discussions regarding solutions and policies.

Of course, addressing this particular issue would require an education element for both policy-makers and cybersecurity engineers who engage in policy matters.

Perhaps a side issue, but one that could greatly influence matters, is that the demonstrations of cyber attacks on physical systems and their impacts can be interpreted as a consequence of the manufacturers or industrial users of those physical systems not being sufficiently sensitive to cybersecurity/safety-related outcomes in their product and system designs. As a result, in carrying out projects, the issue arises regarding reporting on the cybersecurity risks of current systems and the undue reputation impact it could have on the companies whose systems are being used for experimentation. It is not generally understood that the risks are emergent, and that the nature of these findings would be expected across all current software-controlled physical systems that have safety-related outcome potentials. A need exists to address this topic, including defining professional behavior for engineers regarding reporting on the results of their work involving current commercial systems and cyber-attacks and its relationship to the related companies' reputations.

The author of this article has recently served as a Commissioner for Cybersecurity for the Commonwealth of Virginia, which, with strong support from the Governor, has been engaged in strategy development regarding cybersecurity (CoV, 2015). The 11-person Cybersecurity Commission for Virginia, working with Virginia's Cabinet members, has made strong recommendations regarding education programs, and the state has developed budgets to start addressing this need. This state-level initiative is the type of anticipatory action that will be required in order to be prepared should the cyber-attack risks for physical systems materialize.

Cybersecurity Role and Certification of the Operators of Physical Systems

An important aspect of the defense of physical systems from cyber-attacks is that immediate system-reconfiguration responses to attack detections (including what can be very expensive system shut-downs) may be necessary in order to provide the desired level of safety. This calls for doctrine regarding immediate responses. Doctrine must include: 1) the allocation of decision-making and response control roles to specified personnel, 2) selection criteria for, and training of those people, 3) exercising for preparedness, and 4) addressing the possibilities of unanticipated confusion regarding operator judgments related to the possibilities of missed or incorrect attack detections (including zero-day attacks).

Part of the author's research on physical system defense included human involvement in cyber attack scenarios. In the UAV case, a desktop simulation environment was used to gain an initial understanding of operator responses to a monitoring system that detects cyber attacks and provides suggested responses to the UAV pilots. In the State Police case, a controlled exercise was conducted, involving unsuspecting policemen being dispatched, and their cars being attacked and failing to operate properly. The results of these activities highlighted the point that the doctrinal processes to be developed must recognize the fact that cyber attacks on physical systems are an area where people do not and will not have practical experience to rely upon. Furthermore, since attacks are very unlikely to occur, responses may stray from what operators are trained for. The research efforts showed that operators, based on their past experiences, can usually imagine other causes for observed consequences of a cyber attack and, as a result, may not be as responsive to automated decision support as expected.

Consider the case in which a Sentinel detects a cyber-attack that consists of an improper digital control message preventing a car from operating properly. From the operator's perspective there can be many different causes for the car not operating properly (e.g. failed battery), and these are typically causes they have previously experienced. Consequently, under the immediate pressure of needing to take decisive action, the operator may be more likely to assume these causes of failure, rather than a never experienced cyber-attack. Research results showed that even when an operator accepts a Sentinel's input as being correct, uncertainty remains regarding the possibility for additional elements of the cyber-attack having yet to emerge. This element of uncertainty is escalated when there are high consequences associated with an operator's decisions, and the operator's accountability for those decisions can impact behavior, including asking for access to cybersecurity experts before making a critical decision. Of course, such calls for help can potentially delay decision-making to an undesirable degree. As a result of these scenarios actually emerging during our research experiments, a significant effort has been initiated to better understand human behavior in uncertain circumstances that are likely to exist in scenarios regarding cyber-attacks on physical systems. From a policy vantage point, research efforts are needed to address questions regarding selection, certification and readiness training requirements for operators of physical systems for which cyber-attacks could have serious consequences.

Data Curation

Data curation can be defined as the active and ongoing management of data through its lifecycle of interest and usefulness. If one assumes that a critical step in vigorously addressing cybersecurity for physical systems is the need for early evidence that cyber-attacks are actually occurring, significant issues emerge regarding curation of the data that would provide the needed evidence. Based on the automobile-focused State Police project referred to above, an important next step would be the development of accepted policies and processes regarding the collection, storage, security, sharing, analysis, and supplementation of data. For example, consider the case of distribution of specific data that were to be collected at the scene of an automobile incident and, based upon analysis, indicated a possible cyber-attack. Recognizing the international manufacturing base for automobiles and the international sales of automobiles, information would need to be shared across the world. It would be important that worldwide law enforcement agencies, national governments engaged in addressing automobile cybersecurity, automobile companies, and numerous others gain access to that data. As a result, international curation policies and processes would be called for. Organizations such as INTERPOL could potentially play a key role in creating the needed international orientation.

Market Incentives

In February 2014, the National Institute of Standards and Technology (NIST) released Version 1 of White House Executive Order 13636 - Cybersecurity Framework, an initial structure for organizations, government and customers to use in considering comprehensive cybersecurity programs (WH, 2013). In April 2015, a NIST presentation provided a status report on the evolving framework (NIST, 2015). The framework broadly addresses the specific needs that are discussed above, but without the required specificity to illuminate the complexity associated with anticipatory physical system solutions. Past efforts to establish market incentives for improved information system cybersecurity illustrate the consequences of inaction, and also demonstrate the uncertainties and difficulties surrounding anticipatory actions. The example provided by information systems highlights the importance of initiating early data collection efforts so that incidents can be assessed for potential cyber attacks and confirmed attacks can be documented. With this evidence in hand, it will be easier to evaluate next step responses, and incentives for anticipatory forms of cybersecurity will be increased. As emphasized above, it will be difficult to motivate anticipatory solutions without confirmation that attacks on physical systems are actually occurring. The National Highway Safety Traffic System (NHTSA), through guidance that they are providing for improving automobile-related cybersecurity, has taken encouraging steps to anticipate some of the needs addressed above (USDOT, 2016). A potential sequence of events is that data collection starts early and provides incontrovertible evidence of attacks on physical systems, which then drives the development of the needed government, industry and consumer relationships which underpin market incentives for investment in anticipatory cybersecurity. As suggested above, attacks on physical systems generally pose a much greater risk to human safety than attacks on information systems. Therefore, it may be easier to motivate firms and policymakers to invest in physical system security, since potential consequences are so severe. The development of data curation processes that could promote the involvement of appropriate government, industry and consumer groups appears to be a critical early step towards achieving market incentives.

Conclusions and Recommendations

This article emphasizes the point that due to the risk of injuries and deaths associated with cyber-attacks on physical systems, anticipatory cybersecurity solutions are likely to be desired; potentially much more so than has been the case for information system cybersecurity. In addition, a number of examples have been provided that illuminate both the complexity of addressing anticipation and the difficulties associated with selecting and applying the most critical solutions. This complexity includes recognizing the impacts of subsystem interconnections in critical systems, such as air traffic control systems. It has been suggested that managing the implementation of anticipatory solutions will require teams of government and industrial organizations, both to address the consequences of attacks and to design systems for detecting and responding to attacks. The examples highlight the fact that this is an international issue, involving government as well as the relevant industries. The examples also demonstrate that standardization solutions have to consider their monoculture implications in addition to the normal factors that relate to standardization. In order to make progress, our education system needs to prioritize addressing cybersecurity across a broader set of education programs than is currently the practice.

Additionally, it appears likely that evidence of actual cyber attacks on physical systems will be a necessary precursor for anticipatory solutions; due to the associated costs, it is unlikely that self-motivation will be sufficient to drive investment in cybersecurity for physical systems. The creation of market incentives for investment in cybersecurity for physical systems will require the engagement of government, industry and consumer organizations. Since they are first on the scene for incidents of the kind being addressed here, the law enforcement community would seemingly be a logical choice for collecting the needed data. Consequently, the first step in post-event data analysis is equipping law enforcement officers with applicable equipment, so that they can identify events caused by cyber attacks. It is also suggested that industry members engage with the law enforcement community to determine data requirements necessary to identify a cyber attack. Once a number of instances are documented, the policy responses suggested above will likely increase in priority. Hopefully, with appropriate engagement of consumer groups, anticipatory solutions will arise. In order for a rapid response to be possible, an early emphasis must be placed on supporting relevant research and education.

An interesting side note related to this paper is that technology-focused, system prototype experiments served to create early interactions between technologists and policymakers that illuminated a number of important issues related to policy. It would appear that prototype-based projects that serve to couple government and industry would be a valuable method for accelerating the partnerships necessary to identifying and addressing critical policy issues. A preliminary strategy would include identifying safety-related domains that demand the rapid integration of fast changing technologies into their physical systems. This article provides examples related to advanced air traffic control and automated automotive systems.

Acknowledgments

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Systems Engineering Research Center under Contracts HQ0034-13-D-0004. The SERC is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology, Hoboken, NJ, USA. Any opinions, findings, and conclusions or recommendations expressed

in this material are those of the authors and do not necessarily reflect the views of the U.S. Department of Defense.

References

Babineau, G. L., Jones, R. A. and Horowitz, B. M. (2012), A system-aware cyber security method for shipboard control systems with a method described to evaluate cyber security solutions, 2012 IEEE International Conference on Technologies for Homeland Security (HST).

Bayuk, J. L. and Horowitz, B. M. (2011), An architectural systems engineering methodology for addressing cyber security, Systems Engineering 14: 294-304.

Commonwealth of Virginia (CoV) (2015, August), Cyber Security Commission, ”Threats and Opportunities”.

Cyber Security Research Alliance (CART) (2013, April), “Designed-in Cyber Security for Cyber-Physical Systems”, Workshop Report.

Falliere, N., Murchu, L. O. and Chien, E. (2011), “W32.Stuxnet Dossier”, Symantec.

Gay, C. Horowitz, B. Bobko, P., Elshaw, J. & Kim, I. (2017), Operator Suspicion and Decision Responses to Cyber-Attacks on Unmanned Ground Vehicle Systems, HFES 2017 International Annual Meeting, Austin, TX

Higgins, Kelly Jackson, (2015a, September), “State Trooper Vehicles Hacked”, Dark Reading.

Higgins, Kelly Jackson (2015b, July), “Car Hacking Shifts into High Gear” Dark Reading.

Horowitz, B.M. (2016, April), AFCEA SIGNAL – Cybersecurity for Unmanned Aerial Vehicle Missions, pp.40-43.

Horowtiz, B.M. and Pierce, K.M. (2013), The integration of diversely redundant designs, dynamic system models, and state estimation technology to the cyber security of physical systems, Systems Engineering, 16(4): 401-412

Horowitz, B.M., Scott Lucero, D. (2017, September), System-Aware Cybersecurity: A Systems Engineering Approach for Enhancing Cybersecurity, INCOSE INSIGHT, 10.1002/inst.12165

Jones, R.A., Nguyen, T.V. and Horowitz, B.M. (2011), System-Aware security for nuclear power systems, 2011 IEEE International Conference on Technologies for Homeland Security (HST), pp. 224-229.

Jones, R. A. Luckett, B., Beling, P. & Horowitz, B.M. (2013). Architectural Scoring Framework for the Creation and Evaluation of System-Aware Cyber Security Solutions, Journal of Environmental Systems and Decisions 33(3): 341-361.

Jones, R. A., and Horowitz, B. M. (2012). “System-Aware Cyber Security Architecture.” Systems Engineering, February 2012.

Kovacs, Eduard (2014, December), “Cyberattack on German Steel Plant Caused Significant Damage:Report”, Security Week

Miller, Patrick C., (2014a, December), “University of Virginia research protects UAS from cyber-attackers”, UAS Magazine.

Miller, Patrick C. (2014b, December), “Dual Knowledge for UAS Cybersecurity”, UAS Magazine.

NBC29.com (2015, October), “Va. CyberSecurity Research Working to Protect First Responders”, Press Release from the Office of Governor Terry McAuliffe

NIST presentation (2015, April), “Framework for Improving Critical Infrastructure Cybersecurity – Implementation of Executive Order 13636”.

Rieger, C. Gertman, D. & McQueen, M. (2009, May), “Resilient Control Systems: Next Generation Design Research”, International Conference on Human System Interaction.

The White House (WH) (2013, February), “Executive Order – Improving Critical Infrastructure Cybersecurity”.

US DOT (2016) Vissues Federal guidance to the automotive industry for improving motor vehicle security, https://www.nhtsa.gov/press-releases/us-dot-issues-federal-guidance-automotive-industry-improving-motor-vehicle.

On Stream-Centric Learning for Internet of Battlefield Things

Brian Jalaian, Alec Koppel, Andre Harrison,
James Michaelis, Stephen Russell
U.S. Army Research Laboratory
Adelphi, MD, USA
{brian.a.jalaian.civ, alec.e.koppel.civ, andre.v.harrison2.civ,
james.r.michaelis2.civ, stephen.m.russell8.civ}@mail.mil

Abstract

Internet of Things (IoT) technologies have made considerable recent advances in commercial applications, prompting new research on their use in military applications. Towards the development of an Internet of Battlefield Things (IoBT), capable of leveraging mixed commercial and military technologies, several unique challenges of the tactical environment present themselves. These challenges include development of methods for: (I) quickly gathering training data reflecting unforeseen learning/classification tasks; (II) incrementally learning over real-time data streams; (III) management of limited network bandwidth and connectivity between IoBT assets in data gathering and classification tasks. This paper provides a survey over classical and modern statistical learning theory, and how numerical optimization can be used to solve corresponding mathematical problems. The objective of this paper is to encourage the IoT and machine learning research communities to revisit the underlying mathematical underpinnings of stream-based learning, as applicable to IoBT-based systems.

In recent years, Internet of Things (IoT) technologies have seen significant commercial adoption. For IoT technology, a key objective is to deliver intelligent services capable of performing both analytics and reasoning over data streams from heterogeneous device collections. In commercial settings, IoT data processing has commonly been handled through cloud-based services, managed through centralized servers and high-reliability network infrastructures.

Recent advances in IoT technology have motivated the defense community to research IoT architecture development for tactical environments, advancing the development of an Internet of Battlefield Things (IoBT) for use in C4ISR applications (Kott, Swami, and West 2016). Towards advancing IoBT adoption, differences in military vs. commercial network infrastructures become an important consideration. For many commercial IoT architectures, cloud-based services are used to perform needed data processing, which rely upon both stable network coverage and connectivity. As observed in (Zheng and Carter 2015), IoT adoption in the tactical environment faces several technical challenges: (I) Limitations on tactical network connectivity and reliability, which impact the amount of data that can be obtained from IoT sensor collections in real time; (II) Limitations on interoperability between IoT infrastructure components, resulting in reduced infrastructure functionality; (III) Availability of data analytics components accessible over tactical network connections, capable of real-time data ingest over potentially sparse IoT data collections.

Challenges such as these limit the viability of cloud-based service usage in IoBT infrastructures. Hence, significant changes to existing commercial IoT architectures become necessary to ensure their applicability - particularly in the context of machine learning applications. To help illustrate these challenges, a motivating scenario is provided below.

Detecting Vehicle-borne IEDs in Urban Environments:

As part of an ongoing counterinsurgency operation by coalition forces in the country of Aragon, focus is placed on monitoring of insurgent movements and activities. Vehicle-borne IEDs (VBIEDs) have become more frequently used by insurgents in recent months, requiring methods for quick detection and interception. Recent intelligence reports have provided details on physical appearance of IED-outfitted vehicles in the area. However, due to the time constraints in confirming detections of VBIEDs, methods for autonomous detection become desirable. To support VBIED detection, an IoBT infrastructure has been deployed by coalition forces consisting of a mix of Unattended Ground Sensors (UGS) and Unmanned Aerial Systems (UAS). In turn, supervised learning methods are employed over sensor data gathered from both sources.

Recent intelligence has indicated that VBIEDs may be used in a city square during the annual Aragonian Independence Festival. A local custom for this festival involves decoration of vehicles with varying articles (including flags and Christmas tree lighting). A UAS drone is tasked with patrolling airspace over one of the inbound roadways and recoding images of detected vehicles. However, due to the decorations present on many civilian vehicles, confidence in VBIED classification by the UAS is significantly reduced. To mitigate this, the drone flies along a 3 mile stretch of road for 10 minutes to gather new images of the decorated vehicles. In each case, the drone generates a classification of each vehicle as VBIED or not, each with a particular

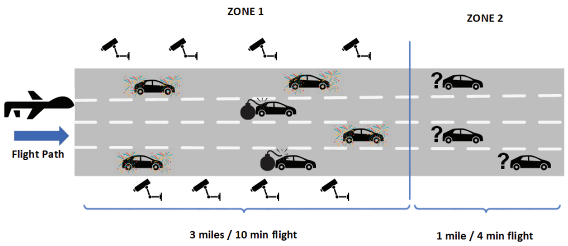


Figure 1: Diagram of Drone Flight over Roadway

confidence value. For low-confidence readings, the drone contacts a corresponding UGS sensor to do the following things: (i) Take a high-resolution image; (ii) Take readings for presence of explosives-related chemicals in air nearby, where any detectable explosives confirms the vehicle is a VBIED. Since battery power for the UGS is limited, along with available network bandwidth, the UAS should only request UGS readings when especially necessary. Following receipt of data from an UGS, the UAS performs retraining of the classifier to improve accuracy of future VBIED classification attempts. Over a short period, the UAS has gathered additional training data to support detection of VBIEDs. Eventually, the drone passes over a 1 mile stretch of road lacking UGS sensors. At this point, the UAS must classify detected vehicles without UGS support.

This example scenario highlights several research issues specific to IoBT settings, as reflected in prior surveys (e.g., (Zheng and Carter 2015; Suri et al. 2016)): (I) a needed capability to quickly gather training data reflecting unforeseen learning/classification tasks; (II) a needed capability to incrementally learn over the stream of field specific data (e.g., increasing the accuracy of classifying VBIEDs by learning over the stream of pictures of decorated cars collected over 10 min of flight time); (III) management of limited network bandwidth and connectivity between assets (e.g., between the UAS and UGS along the road) requiring selective asset use to obtain classifier relevant data that increases the classifier knowledge;

Each of these issues require the selection of learning and classification methods appropriate to stream-based data sources. Prior research (Bottou 1998b) (Vapnik 2013), (Bottou and Cun 2004) demonstrates the equivalence of learning from stream-based data in real time with learning from *infinitely* many samples. From this work, it follows that statistical learning methods adept to large-scale data sources may be applicable for stream-based data.

This paper opens with a survey over classical and modern statistical learning theory, and how numerical optimization can be used to solve the resulting mathematical problems. The objective of this paper is to encourage the IoT and machine learning research communities to revisit the underlying mathematical underpinnings of stream-based learning, as applicable to IoBT-based systems.

Statistical Learning and Stochastic Optimization

In statistical learning, we are given data in the form of independent and identically distributed (i.i.d.) samples $\mathbf{x}_n$ of a random variable $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^p$. Based upon $\mathbf{x}_n$ we would like to estimate some target variable $\mathbf{y}_n$ which is an i.i.d. sample random variable $\mathbf{y} \in \mathcal{Y}$. $\mathcal{X}$ is typically called the feature space, and $\mathcal{Y}$ is called the target domain, which may be a discrete set $\{1, \ldots, C\}$ in the case of classification or $\mathcal{Y} \subset \mathbb{R}^q$ in the case of regression. Ideally, one would like to select an estimator $\hat{\mathbf{y}}(\mathbf{x})$ which makes the minimal number of mistakes in expectation over all data, also known as the *statistical error rate*:

$$\tilde{\mathbf{y}}^\star = \operatorname*{argmin}_{\hat{\mathbf{y}} \in \mathcal{Y}^\mathcal{X}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\mathbb{1}\{\hat{\mathbf{y}}(\mathbf{x}) \neq \mathbf{y}\}] \tag{1}$$

Here we use $\mathcal{Y}^\mathcal{X}$ to denote the space of all functions from feature space $\mathcal{X}$ to target domain $\mathcal{Y}$. There are two fundamental issues in minimizing the statistical error (1) which make it intractable: it is NP-hard to optimize over an integer-valued stochastic function and that the feasible set, when a generic function space without any structure, is mathematically impossible to optimize over (Murty and Kabadi 1987). Researchers have addressed the later issue in a variety of ways, leading to the rich field of supervised machine learning; on the other hand, a unifying thread of these approaches is the approximation of the indicator in (1) by a convex loss function $\ell :\in \mathcal{Y}^\mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ which is small when $\hat{\mathbf{y}}(\mathbf{x})$ is close to $\mathbf{y}$ and large when far apart. Doing so then yields the General Learning setting of (Vapnik 1995)

$$\mathbf{y}^\star = \operatorname*{argmin}_{\hat{\mathbf{y}} \in \mathcal{Y}^\mathcal{X}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(\hat{\mathbf{y}}(\mathbf{x}), \mathbf{y})] \tag{2}$$

The solution to the General Learning setting for the arbitrarily complicated feasible set $\mathcal{Y}^\mathcal{X}$ is denoted as the Bayes optimal, and the function to the right of the equality in (2) is called the Bayes risk (Hastie, Tibshirani, and Friedman 2009). Before surveying different functional specifications $\mathcal{F}$ in lieu of $\mathcal{Y}^\mathcal{X}$, we recall the *bias-variance decomposition*, notions of statistical consistency, how these motivate use of increasingly complicated choices of $\mathcal{F}$ when more data is available.

First, observe that to compute the Bayes optimal, we need to compute an expectation over the unknown joint distribution of $(\mathbf{x}, \mathbf{y})$ which is impossible. In practice, we have access to N i.i.d. training examples $(\mathbf{x}_n, \mathbf{y}_n)$, and can compute some estimate $\hat{\mathbf{y}}_N$. For instance, $\hat{\mathbf{y}}_N$ could be the minimizer of the empirical risk, stated as $(1/N)\sum_n \ell(\hat{\mathbf{y}}(\mathbf{x}_n), \mathbf{y}_n)$, or N iterations of stochastic gradient method. The cost-difference between $\hat{\mathbf{y}}_N$ and the Bayes optimal $\mathbf{y}^\star$ is

$$\begin{aligned} &\mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(\hat{\mathbf{y}}_N(\mathbf{x}), \mathbf{y})] - \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(\hat{\mathbf{y}}^\star(\mathbf{x}), \mathbf{y})] \\ &\quad = \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(\hat{\mathbf{y}}_N(\mathbf{x}), \mathbf{y})] - \min_{\hat{\mathbf{y}} \in \mathcal{F}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(\hat{\mathbf{y}}(\mathbf{x}), \mathbf{y})] \\ &\qquad + \min_{\hat{\mathbf{y}} \in \mathcal{F}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(\hat{\mathbf{y}}_N(\mathbf{x}), \mathbf{y})] - \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(\hat{\mathbf{y}}^\star(\mathbf{x}), \mathbf{y})] \end{aligned} \tag{3}$$

In (3), we have added and subtracted $\min_{\hat{\mathbf{y}} \in \mathcal{F}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(\hat{\mathbf{y}}_N(\mathbf{x}), \mathbf{y})]$, the cost associated with optimal estimator within our hypothesized function class $\mathcal{F}$.

Observe that the first two terms on the right-hand side of (3), called the *model bias* or estimation error, may be made small by increasing N, the sample size of our training data.

On the other hand, the later two terms, called the *model variance*, or approximation error, is a fixed function of our modeling hypothesis in the form of our choice of function class $\mathcal{F}$. This suggests to just use arbitrarily complicated choices of $\mathcal{F}$; however, for a fixed N, the estimation error *increases* as the complexity of $\mathcal{F}$ increases. This is because the difference between the minimizer of the empirical and expected risk has been established to be at least proportional to $\mathcal{O}(|\mathcal{F}|/\sqrt{N})$ (Castro 2015)[Ch. 7, Prop. 3] or (Hastie, Tibshirani, and Friedman 2009). The fundamental trade off of the complexity of our modeling hypothesis $\mathcal{F}$ with sample size N, known colloqially as the bias-variance decomposition, gives rise to the field of structured risk minimization (Shawe-Taylor et al. 1998) and model selection (Bartlett, Boucheron, and Lugosi 2002; Koltchinskii and others 2009).

Rather than emphasize this tradeoff further, we discuss whether statistical consistency is attainable within a given function class, and how this motivates different learning techniques. By statistical consistency for fixed $\mathcal{F}$, we mean

$$\lim_{N\to\infty} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(\hat{\mathbf{y}}_N(\mathbf{x}),\mathbf{y})] = \min_{\hat{\mathbf{y}}\in\mathcal{F}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(\hat{\mathbf{y}}(\mathbf{x}),\mathbf{y})] \ . \quad (4)$$

If our choice of $\mathcal{F}$ is general enough that it contains the optimizer $\mathbf{y}^\star$ within $\mathcal{Y}^\mathcal{X}$, then we have solved the problem. Unfortunately, in practice, one never knows. Thus, we adopt an engineering approach in which we discuss various choices of $\mathcal{F}$ in order of progressively increasing complexity, the attainability of consistency (optimality) for that function class, and numerical tools for attaining these statistical optimizers.

Generalized Linear Models (GLMs) The first and simplest choice of estimator function class is $\mathcal{F} = \mathbb{R}^p$. In this case, the estimator is a generalized linear model (GLM): $\hat{\mathbf{y}}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$ for some parameter vector $\mathbf{w} \in \mathbb{R}^p$ (Nelder and Baker 1972). In this case, optimizing the statistical loss is the stochastic convex optimization problem, stated as

$$\min_{\mathbf{w}\in\mathbb{R}^p} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(\mathbf{w}^T\mathbf{x},\mathbf{y})] \ . \quad (5)$$

Observe that to optimize (5), assuming a closed form solution is unavailable, must be done using gradient descent or Newton's method (Boyd and Vanderberghe 2004). However, either method requires computing the gradient of $L(\mathbf{w}) := \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(\mathbf{w}^T\mathbf{x},\mathbf{y})]$ which requires infinitely many realizations $(\mathbf{x}_n,\mathbf{y}_n)$ of the random pair $(\mathbf{x},\mathbf{y})$, and thus has infinite complexity. This computational bottleneck has been resolved through the development of stochastic approximation (SA) methods (Robbins and Monro 1951; Bottou 1998a) which operate on subsets of data examples per step. The most common SA method is the stochastic gradient method (SGD), which involves descending along the stochastic gradient $\nabla_\mathbf{w}\ell(\mathbf{w}^T\mathbf{x}_t,\mathbf{y}_t)$ rather than the true gradient at each step:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t\nabla_\mathbf{w}\ell(\mathbf{w}_t^T\mathbf{x}_t,\mathbf{y}_t) \quad (6)$$

Use of SGD (6) is prevalent due to its simplicity, ease of use, and the fact that it converges to the minimizer of (5) almost surely, and in expectation at a $\mathcal{O}(1/\sqrt{t})$ rate when $L(\mathbf{w})$ is convex and a sublinearly $\mathcal{O}(1/t)$ when it is strongly convex. Efforts to improve the mean convergence rate to $\mathcal{O}(1/t^2)$ through the use of Nesterov acceleration (Nemirovski et al. 2009) have also been developed, whose updates are given as

$$\begin{aligned}\mathbf{w}_{t+1} &= \mathbf{v}_t - \eta_t\nabla_\mathbf{w}\ell(\mathbf{v}_t^T\mathbf{x}_t,\mathbf{y}_t)\\ \mathbf{v}_{t+1} &= (1-\gamma_t)\mathbf{w}_{t+1} + \gamma_t\mathbf{w}_t\end{aligned} \quad (7)$$

A plethora of tools have been proposed specifically to minimize the empirical risk (sample size N is finite) in the case of GLMs, which achieve even faster *linear* or superlinear convergence rates. These methods are either based on reducing the variance of the stochastic approximation (data-subsampling) error of the stochastic gradient (Schmidt, Roux, and Bach 2013; Johnson and Zhang 2013; Defazio, Bach, and Lacoste-Julien 2014) or by using approximate second-order (Hessian) information (Goldfarb 1970; Shanno and Phua 1976). This thread has culminated in the fact that Quasi-Newton methods (Mokhtari, Gürbüzbalaban, and Ribeiro 2016) outperform variance reduction methods (Hu, Pan, and Kwok 2009) for finite-sum minimization when N is large-scale. For specifics on stochastic Quasi-Newton updates, see (Mokhtari and Ribeiro 2015; Byrd et al. 2016). However, as $N \to \infty$, the analysis which yields linear or superlinear learning rates breaks down, and the best one can hope for is Nesterov's $\mathcal{O}(1/t^2)$ rate (Nemirovski et al. 2009).

Learning Feature Representations for Inference Transformations of data domains have become widely used in the past decades, due to their ability to extract useful information from input signals as a precursor to solving statistical inference problems. For instance, if the signal dimension is very large, dimensionality reduction is of interest, which may be approached with principal component analysis (Jolliffe 1986). If instead one would like to conduct multi-resolution analysis, wavelets (Mallat 2008) may be more appropriate. These techniques, which also include as k-nearest neighbor, are known as unsupervised or signal representation learning (Murphy 2012). Recently, methods based on learned representations, rather than those fixed a priori, have gained traction in pattern recognition (Elad and Aharon 2006; Mairal, Elad, and Sapiro 2007). A special case of data-driven representation learning is dictionary learning (Mairal et al. 2008), the focus of this sub-section.

Here we address finding a dictionary (signal encoding) that is well adapted to a specific inference task (Mairal, Bach, and Ponce 2012). To do so, denote the coding $\boldsymbol{\alpha}(\tilde{\mathbf{D}};\mathbf{x}) \in \mathbb{R}^k$ as a feature representation of the signal $\mathbf{x}_t$ with respect to some dictionary matrix $\tilde{\mathbf{D}} \in \mathbb{R}^{p\times k}$. Typically, $\boldsymbol{\alpha}(\tilde{\mathbf{D}};\mathbf{x})$ is chosen as the solution to a lasso regression or approximate solution to an ℓ_0 constrained problem that minimizes some criterion of distance between $\mathbf{D}^T\boldsymbol{\alpha}$ and $\mathbf{x}$ to incentivize codes to be sparse. Further introduce the classifier $\mathbf{w} \in \mathbb{R}^k$ that is used to predict target variable $\mathbf{y}_t$ when given the signal encoding $\boldsymbol{\alpha}(\tilde{\mathbf{D}};\mathbf{x})$. The merit of the classifier $\mathbf{w} \in \mathcal{W} \subset \mathbb{R}^k$ is measured by the smooth loss function $\ell\big(\boldsymbol{\alpha}^*(\mathbf{w}^T\boldsymbol{\alpha}(\tilde{\mathbf{D}};\mathbf{x});(\mathbf{x}_t,\mathbf{y}_t)\big)$ that captures how well the classifier $\mathbf{w}$ may predict $\mathbf{y}_t$ when given the coding $\boldsymbol{\alpha}(\tilde{\mathbf{D}};\mathbf{x}_t)$. Note that α is computed using the dictionary $\tilde{\mathbf{D}}$. The task-driven dictionary learning problem is formulated as the joint determination of the dictionary $\tilde{\mathbf{D}} \in \mathcal{D}$ and classifier $\mathbf{w} \in \mathcal{W} \subset \mathbb{R}^k$ that minimize the cost $\ell\big(\boldsymbol{\alpha}(\tilde{\mathbf{D}};\mathbf{x}_t),\mathbf{w};(\mathbf{x}_t,\mathbf{y}_t)\big)$ averaged over the training set,

$$(\tilde{\mathbf{D}}^*,\mathbf{w}^*) := \underset{\tilde{\mathbf{D}}\in\mathcal{D},\mathbf{w}\in\mathcal{W}}{\operatorname{argmin}}\ \mathbb{E}_{\mathbf{x},\mathbf{y}}\Big[\ell\big(\mathbf{w}^T\boldsymbol{\alpha}(\tilde{\mathbf{D}};\mathbf{x}_t);(\mathbf{x}_t,\mathbf{y}_t)\big).\Big] \tag{8}$$

In (8), we specify the estimator $\hat{\mathbf{y}}(\mathbf{x}) = \mathbf{w}^T\boldsymbol{\alpha}^*(\tilde{\mathbf{D}};\mathbf{x})$, which parameterizes the function class $\mathcal{F}$ as the product set $\mathcal{W}\times\mathcal{D}$. For a given dictionary $\tilde{\mathbf{D}}$ and signal sample $\mathbf{x}_t$ we compute the code $\boldsymbol{\alpha}^*(\tilde{\mathbf{D}};\mathbf{x}_t)$ as per some lasso regression problem, for instance, and then predict $\mathbf{y}_t$ using $\mathbf{w}$, and measure the prediction error with the loss function $\ell\big(\mathbf{w}^T\boldsymbol{\alpha}(\tilde{\mathbf{D}};\mathbf{x}_t),;(\mathbf{x}_t,\mathbf{y}_t)\big)$. The optimal pair $(\tilde{\mathbf{D}}^*,\mathbf{w}^*)$ in (8) is the one that minimizes the cost averaged over the given sample pairs $(\mathbf{x}_t,\mathbf{y}_t)$. Observe that $\boldsymbol{\alpha}^*(\tilde{\mathbf{D}};\mathbf{x}_t)$ is not a variable in the optimization in (8) but a mapping for an implicit dependence of the loss on the dictionary $\tilde{\mathbf{D}}$. The optimization problem in (8) is not assumed to be convex – this would be restrictive because the dependence of ℓ on $\tilde{\mathbf{D}}$ is, partly, through the mapping $\boldsymbol{\alpha}^*(\tilde{\mathbf{D}};\mathbf{x}_t)$ defined by some sparse-coding procedure. In general, only local minima of (8) can be found. This formulation has nonetheless been successful in solving practical pattern recognition tasks in vision (Mairal, Bach, and Ponce 2012) and robotics (Koppel et al. 2016a).

The lack of convexity of (8) means that attaining statistical consistency [cf. (4)] for supervised dictionary learning methods is much more challenging than for GLMs. To this end, the prevalence of non-convex stochastic programs arising from statistical learning based on nonlinear transformations of the feature space $\mathcal{X}$ has led to a renewed interest in non-convex optimization methods through applying convex techniques to non-convex settings (Boyd and Vanderberghe 2004). This constitutes a form of simulated annealing (Bertsimas and Tsitsiklis 1993) with successive convex approximation (Facchinei, Scutari, and Sagratella 2015). A compelling achievement of this recent surge is the hybrid convex-annealing approach which has been shown to be capable of finding a global minimizer (Raginsky, Rakhlin, and Telgarsky 2017). However, the use of these methods for addressing the training of estimators defined by non-convex stochastic programs requires far more training examples to obtain convergence than convex problems, and requires further demonstration in practice.

Reproducing Kernel Hilbert Spaces (RKHS) Now we shift focus to the case where $\mathcal{F}$ is not a p-dimensional vector space but is instead a generic Hilbert space $\mathcal{H}$ equipped with an inner-product-like function called a *reproducing kernel* (Kimeldorf and Wahba 1971). The reason for this specification is that in practice one obtains much smaller approximation errors when selecting more expressive choices of $\mathcal{F}$, and this selection crucially allows for the learning of nonlinear statistical models while preserving convexity. For this case, the statistical loss takes the form

$$\min_{f\in\mathcal{H}} \mathbb{E}_{\mathbf{x},\mathbf{y}}[\ell(f(\mathbf{x}),\mathbf{y})] + \frac{\lambda}{2}\|f\|_{\mathcal{H}}\ . \tag{9}$$

λ is a regularization parameter which ensures (9) is strongly convex. Here the kernel associated with $\mathcal{H}$ is defined over the product feature space, i.e., $\kappa : \mathcal{X}\times\mathcal{X}\to\mathbb{R}$, such that elements of $\mathcal{H}$ are *functions*, $f:\mathcal{X}\to\mathcal{Y}$ which satisfy

$$\begin{aligned} &(i)\ \langle f,\kappa(\mathbf{x},\cdot))\rangle_{\mathcal{H}} = f(\mathbf{x}) \quad \text{for all } \mathbf{x}\in\mathcal{X}\ , \\ &(ii)\ \mathcal{H} = \overline{\operatorname{span}\{\kappa(\mathbf{x},\cdot)\}} \quad \text{for all } \mathbf{x}\in\mathcal{X}\ . \end{aligned} \tag{10}$$

where $\langle\cdot,\cdot\rangle_{\mathcal{H}}$ denotes the Hilbert inner product for $\mathcal{H}$. When the kernel is positive semidefinite, i.e. $\kappa(\mathbf{x},\mathbf{x}')\geq 0$ for all $\mathbf{x},\mathbf{x}'\in\mathcal{X}$, this function space is called a reproducing kernel Hilbert space (RKHS).

In (10), property *(i)* is called the reproducing property of the kernel, and is a consequence of the Riesz Representation Theorem (Wheeden, Wheeden, and Zygmund 1977). Replacing f by $\kappa(\mathbf{x}',\cdot)$ in (10) (i) yields the expression $\langle\kappa(\mathbf{x}',\cdot),\kappa(\mathbf{x},\cdot)\rangle_{\mathcal{H}} = \kappa(\mathbf{x},\mathbf{x}')$, which is the origin of the term “reproducing kernel.” This property provides a practical means by which to access a nonlinear transformation of the input space $\mathcal{X}$. Specifically, denote by $\phi(\cdot)$ a nonlinear map of the feature space that assigns to each $\mathbf{x}$ the kernel function $\kappa(\cdot,\mathbf{x})$. Then the reproducing property of the kernel allows us to write the inner product of the image of distinct feature vectors $\mathbf{x}$ and $\mathbf{x}'$ under the map ϕ in terms of kernel evaluations only: $\langle\phi(\mathbf{x}),\phi(\mathbf{x}')\rangle_{\mathcal{H}} = \kappa(\mathbf{x},\mathbf{x}')$. This is commonly referred to as the *kernel trick*, and it provides a tool for learning nonlinear functions.

Moreover, property (10) *(ii)* states that any function $f\in\mathcal{H}$ may be written as a linear combination of kernel evaluations. For kernelized and regularized empirical risk minimization, the Representer Theorem (Kimeldorf and Wahba 1971; Schölkopf, Herbrich, and Smola 2001) establishes that the optimal f in the hypothesis function class $\mathcal{H}$ may be written as an expansion of kernel evaluations *only* at elements of the training set as

$$f(\mathbf{x}) = \sum_{n=1}^{N} w_n\kappa(\mathbf{x}_n,\mathbf{x})\ . \tag{11}$$

where $\mathbf{w} = [w_1,\cdots,w_N]^T \in \mathbb{R}^N$ denotes a set of weights. The upper summand index N in (11) is henceforth referred to as the model order. Common choices κ include the polynomial kernel and the radial basis kernel, i.e., $\kappa(\mathbf{x},\mathbf{x}') = \big(\mathbf{x}^T\mathbf{x}'+b\big)^c$ and $\kappa(\mathbf{x},\mathbf{x}') = \exp\Big\{-\frac{\|\mathbf{x}-\mathbf{x}'\|_2^2}{2c^2}\Big\}$, respectively, where $\mathbf{x},\mathbf{x}'\in\mathcal{X}$. Unfortunately, as sample size $N\to\infty$, the function representation requires infinite complexity as well (Norkin and Keyzer 2009). This issue is called the ”curse of kernelization.”

Compounding this issue is the fact that when one derives the functional generalization of a stochastic gradient method (Kivinen, Smola, and Williamson 2004; Koppel et al. 2016b), stated as

$$\begin{aligned} f_{t+1} &= (1-\eta_t\lambda)f_t - \eta_t\nabla_f\ell(f_t(\mathbf{x}_t),y_t) \\ &= (1-\eta_t\lambda)f_t - \eta_t\ell'(f_t(\mathbf{x}_t),y_t)\kappa(\mathbf{x}_t,\cdot)\ , \end{aligned} \tag{12}$$

the complexity of storing f_t is $\mathcal{O}(t)$ due to the fact that each stochastic gradient step centers a kernel dictionary element $\kappa(\mathbf{x}_t, \cdot)$ at the latest training example $\mathbf{x}_t$.

Efforts to mitigate "the curse of kernelization" have been previously developed. These combine functional extensions of stochastic gradient method (12) with compressions of the function parameterization independently of the optimization problem to which they are applied (Engel, Mannor, and Meir 2004; Liu, Pokharel, and Principe 2008; Kivinen, Smola, and Williamson 2004; Dekel, Shalev-Shwartz, and Singer 2006; Zhu and Hastie 2005; Richard, Bermudez, and Honeine 2009) or approximate the kernel during training (Dai et al. 2014; Le et al. 2016b; 2016a; Lu et al. 2016), and at best converge on average. In contrast, a method was recently proposed that combines greedily constructed (Pati, Rezaiifar, and Krishnaprasad 1993) sparse subspace projections with a functional stochastic gradient method and guarantees exact convergence to the minimizer of the average risk functional. This technique, called parsimonious online learning with kernels (POLK), tailors the parameterization compression to preserve the descent properties of the underlying RKHS-valued stochastic process (Koppel et al. 2016b). The update of POLK is given as

$$\begin{aligned}\tilde{f}_{t+1}(\cdot) &= (1-\eta_t\lambda)f_t - \eta_t\ell'(f_t(\mathbf{x}_t), \mathbf{y}_t)\kappa(\mathbf{x}_t, \cdot)\\ (f_{t+1}, \mathbf{D}_{t+1}, \mathbf{w}_{t+1}) &= \textbf{KOMP}(\tilde{f}_{t+1}, \tilde{\mathbf{D}}_{t+1}, \tilde{\mathbf{w}}_{t+1}, \epsilon_t) \quad (13)\end{aligned}$$

where between the first and second lines above (13), we define the updates for the dictionary $\tilde{\mathbf{D}}_{t+1} = [\mathbf{D}_t, \quad \mathbf{x}_t]$ and weights $\tilde{\mathbf{w}}_{t+1} \leftarrow [(1-\eta_t\lambda)\mathbf{w}_t, \quad -\eta_t\ell'(f_t(\mathbf{x}_t), y_t)]$ as those arising from the functional stochastic gradient method (12). For details of how matching pursuit is applied, see (Pati, Rezaiifar, and Krishnaprasad 1993; Koppel et al. 2016b).

When the compression budget and learning rates are constant, i.e., $\epsilon_t = \epsilon$ and $\eta_t = \eta$ such that $\epsilon = \mathcal{O}(\eta^{3/2})$, it is possible to find the optimally sparse finite-memory regression function through only sequentially revealed i.i.d. training examples $(\mathbf{x}_t, \mathbf{y}_t)$ – see (Koppel et al. 2016b)[Theorems 2 - 3]. However, since this is a first-order stochastic method, its (not-yet-derived) learning rate will be at best sublinear $\mathcal{O}(1/t)$ in expectation for positive regularizer $\lambda > 0$ (which ensures strong convexity). It is an open question whether these rates can be improved by RKHS extensions of stochastic Nesterov acceleration or stochastic Quasi-Newton techniques, although a recent effort to develop the later approach has appeared for a related setting (Calandriello, Lazaric, and Valko 2017). Moreover, efforts to use increasingly complicated convolutional and hierarchical kernels have recently appeared (Mairal et al. 2014; Mairal 2016) which attempt to encapsulate the multi-resolution properties of wavelets (Mallat 2008) and deep learners (Haykin 1994). However, their use in statistical learning for obtaining statistical consistency with hierarchical kernels is yet not well-understood.

Neural Networks While the mathematical formulation of convolutional neural networks and their variants have been around for decades (Haykin 1994), their use has only become widespread in recent years as computing power and data pervasiveness has made them not impossible to train. Since the landmark work (Krizhevsky, Sutskever, and Hinton 2012) demonstrated their ability to solve image recognition tasks on much larger scales than previously addressable, they have permeated many fields such as speech (Graves, Mohamed, and Hinton 2013), text (Jaderberg et al. 2016), and control (Lillicrap et al. 2015). Rather than review their achievements, we focus on their mathematical formulation, how they relate to the General Learning setting (2), and what is known about their training. That is, consider (2) with the estimator function class $\mathcal{F}$ being defined by the composition of many functions of the form $g_k(\mathbf{x}) = \mathbf{w}_k\sigma_k(\mathbf{x})$. Here σ_k is a nonlinear "activation function" which can be, e.g., a rectified linear unit $\sigma_k(a) = \max(a, 0)$, a sigmoid $\sigma_k(a) = 1/(1+e^a)$, or a hyperbolic tangent $\sigma_k(a) = (1-e^{-2a})/(1+e^{-2a})$. Specifically, for a K-layer convolutional neural network, the estimator is given as

$$\hat{\mathbf{y}}(\mathbf{x}) = g_1 \circ g_2 \circ \cdots g_K(\mathbf{x}) \quad (14)$$

and typically one tries to make the distance between the target variable and the estimator small by minimizing their quadratic distance

$$\min_{\mathbf{w}_1,\ldots,\mathbf{w}_K} \mathbb{E}_{\mathbf{x},\mathbf{y}}(\mathbf{y} - g_1 \circ g_2 \circ \cdots g_K(\mathbf{x}))^2 \quad (15)$$

where each $\mathbf{w}_k$ is a vector whose length depends on the number of "neurons" at each layer of the network. This operation may be thought of as an iterated generalization of a convolutional filter. Additional complexities can be added at each layer, such as aggregating values output for the activation functions by their maximum (max pooling) or average. But the training procedure is similar: to minimize a variant of the highly non-convex, high-dimensional stochastic program (15). Due to their high dimensionality, efforts to modify non-convex stochastic optimization algorithms to be amenable to parallel computing architectures have gained salience in recent years. An active area of research is the interplay between parallel stochastic algorithms and scientific computing to minimize the clock time required for training neural networks – see (Lian et al. 2015; Mokhtari et al. 2017; Scardapane and Di Lorenzo 2017), for instance. Thus far, efforts have been restricted to attaining computational speedup by parallelization to convergence at a stationary point, although some preliminary efforts to escape saddle points and ensure convergence to a local minimizer have also recently appeared (Lee et al. 2016); these modify convex optimization techniques, for instance, by replacing indefinite Hessians with positive definite approximate Hessians (Paternain, Mokhtari, and Ribeiro 2017).

Incremental learning

In many statistical learning problems it is common to assume that the statistical properties of the input data and the output data are stationary. Once a model is trained on a sufficient number of samples to achieve a certain level of performance no further training is therefore needed. However, there are many situations where these assumptions at not valid. 1.) All of the training data is not available or it is not desirable to use all of the training data at once. In this case a model must trained using an incomplete set of data. 2.) The statistical properties of the input may change. 3.) The relationship between the input data and the target classes may change. This is known as concept drift.

Incremental learning provides a way to address all of these cases without having to store all of the data that has been seen and without having to retrain an entirely new model. A previously trained model can simply be updated with new data, either each time a new data sample is received or when blocks of data are received. By incrementally updating trained models these models can efficiently adapt to changing scenarios or requirements. Training on smaller blocks of data may also ease the computational complexity of the training step.

In the resource constrained environment of IOBT systems this efficient use of data and resources can be very valuable. For many statistical batch learning methods there exist variations to those approaches that allow them to be trained incrementally (Ruping 2001) (Diehl and Cauwenberghs 2003) (Jurafsky and Martin 2017), (Cauwenberghs and Poggio), (Agrawal and Bala 2008), (Huang et al. 2015), (Zang et al. 2014). However, the resource constrained nature of IOBT systems encourages the use of simpler or more linear learning models, which we review in the following subsections. We also focus on supervised incremental learning vs. unsupervised or semi-supervised approaches.

When training on blocks of data we define each block of data as $(\mathbf{X}_i, \mathbf{Y}_i) = \mathbf{D}_i$, where each block consists of T training examples $(\mathbf{x}_{i,j}, \mathbf{y}_{i,j})$. For each block we try to train an optimal predictor $\mathbf{y}^{\star}_{1:i}$ by combining the prior optimal estimator $\mathbf{y}^{\star}_{i:i-1}$ with the improvements given by the training set $(\mathbf{X}_i, \mathbf{Y}_i)$.

Naïve Bayes

The standard Naïve Bayes calculation to find the optimal mapping of $\mathbf{Y}^{\star}\colon \mathbf{X} \mapsto \mathcal{Y}$ using an estimator of Y is given by:

$$\hat{\mathbf{y}}_{1:i}(\mathbf{X}) = P_{1:i}(\mathbf{Y}|\mathbf{X}) = \frac{P_{1:i}(\mathbf{X}|\mathbf{Y})P_{1:i}(\mathbf{Y})}{P_{1:i}(\mathbf{X})} \tag{16}$$

In the case that X and y are both discrete we try to predict the probability that $\mathbf{Y}_i = c_k$ and $c_k \in C = \{c_1, c_2, ..., c_L\}$ (Zang et al. 2014). In which case the prior probability is

$$P_{1:i}(\mathbf{Y} = c_k) = \frac{1 + \mathit{count}(\mathbf{y}_{1:i} = c_k)}{L + iT} \tag{17}$$

and the likelihood probability is

$$P_{1:i}(\mathbf{X}_{1:i}|\mathbf{Y}_{1:i} = c_k) = \frac{1 + \mathit{count}(\mathbf{X}_{1:i} \cap c_k)}{|\mathbf{X}_{1:i}| + \mathit{count}(c_k)} \tag{18}$$

Where $|\mathbf{X}_{1:i}|$ is the number of unique values of $\mathbf{X}|$. The prior probability and likelihood estimates are incremented when a new set of training samples are received $(\mathbf{X}_{i+1}, \mathbf{Y}_{i+1})$. The new prior probability is then

$$P_{1:i}(\mathbf{Y} = c_k) = \frac{1 + \mathit{count}(c_k) + \mathit{count}'(c_k)}{L + iT + T} \tag{19}$$

and the new likelihood probability is

$$P_{1:i}(\mathbf{X}|\mathbf{Y} = c_k) = \frac{1 + \mathit{count}(\mathbf{X}_{1:i} \cap c_k) + \mathit{count}(\mathbf{X}_{i+1} \cap c_k)}{|\mathbf{x}_{1:i}| + \mathit{count}(c_k) + \mathit{count}'(c_k)} \tag{20}$$

If $\mathbf{X}$ is continuous then the conditional probability can be modeled as a Gaussian function

$$P(\mathbf{x}|\mathbf{y} = c_k) = \frac{1}{\sigma_{i,k}\sqrt{2\pi}} \exp \frac{-(\mathbf{x} - \mu_{i,k})^2}{2\sigma^2_{i,k}} \tag{21}$$

where μ and σ can be calculated using maximum likelihood estimates

$$\begin{aligned} \hat{\mu}_{i,k} &= \frac{\sum_j \mathbf{x}_{i,j}\delta_{c_k}(\mathbf{y}_{i,j})}{\sum_j \delta_{c_K}(\mathbf{y}_{i,j})} \\ &= \frac{\mathbf{S}_{c_k}(\mathbf{x}, \mathbf{y})}{\mathit{count}_i(c_k)} \end{aligned} \tag{22}$$

$$\hat{\mu}_{i+1,k} = \frac{\mathbf{S}_{c_k}(\mathbf{x}, \mathbf{y}) + \sum_j \mathbf{x}_{i+1,j}\delta_{c_k}(\mathbf{y}_{i+1,j})}{\mathit{count}_i(c_k) + \sum_j \delta_{c_k}(\mathbf{y}_{i+1,j})} \tag{23}$$

and

$$\begin{aligned} \hat{\sigma}^2_{i,k} &= \frac{\sum_j (\mathbf{x}_{i,j} - \hat{\mu}_{i,k})^2 \delta_{c_k}(\mathbf{y}_{i,j})}{\sum_j \delta_{c_k}(\mathbf{y}_{i,j}) - 1} \\ &= \frac{\mathbf{S}_{c_k}(\mathbf{x}^2, \mathbf{y}) + 2\hat{\mu}_{i,k}\mathbf{S}_{c_k}(\mathbf{x}, \mathbf{y}) + T\hat{\mu}^2_{i,k}}{\mathit{count}_i(c_k) - 1} \end{aligned} \tag{24}$$

$$\begin{aligned} \hat{\sigma}^2_{i+1,k} = \frac{1}{\mathit{count}_i(c_k) + \sum_j \delta_{c_k}(\mathbf{y}_{i+1,j}) - 1} \Big(\mathbf{S}_{c_k}(\mathbf{x}^2, \mathbf{y}) + \\ 2(\hat{\mu}_{i,k} + \triangle\hat{\mu}_{i,k})\mathbf{S}_{c_k}(\mathbf{x}, \mathbf{y}) + T(\hat{\mu}_{i,k} + \triangle\hat{\mu}_{i,k})^2 \Big) \end{aligned} \tag{25}$$

Where $\mu_{i+1} = (\mu_i + \triangle\mu_i)$. The Gaussian Naïve Bayes approach is a very popular approach to modeling the likelihood function of the Naïve Bayes algorithm (Metsis, Androutsopoulos, and Paliouras 2006; Losing, Hammer, and Wersing 2018), however other distributions have been used to model the likelihood function (Metsis, Androutsopoulos, and Paliouras 2006; Anderson and Matessa 1992).

Extreme Learning Machines

Are feed-forward neural networks typically configured as a perceptron with a single hidden layer. The output of the network can be described by the equation:

$$\hat{\mathbf{Y}}(\mathbf{x}) = \sum_{p=1}^{L} \beta_p h_p(\mathbf{x}) \tag{26}$$

where $\boldsymbol{\beta} = [\beta_1, \beta_2, ..., \beta_L]$ -]are the weights between the output node and the L hidden layer nodes and $h(\mathbf{x})$ is the non-linear feature mapping between input $\mathbf{x}$ and the hidden layer. One of the unique features of ELM networks is that rather than training each $h_i(\mathbf{x})$ function they are created randomly with random parameter values, typically from piecewise non-linear continuous distribution functions. Training the ELM network simply requires optimizing the mapping from the hidden layer to the output.

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^{\mathbf{LXM}}} ||\mathbf{H}\boldsymbol{\beta} - \mathbf{T}||^2 \tag{27}$$

where $\mathbf{H}$ is the set hidden layer functions applied to the inputs of the ELM network

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_1(\mathbf{x}_1) \\ \mathbf{h}_2(\mathbf{x}_2) \\ \vdots \\ \mathbf{h}_L(\mathbf{x}_N) \end{bmatrix} = \begin{bmatrix} h_1(\mathbf{x}_1) & h_2(\mathbf{x}_1) & \dots & h_L\mathbf{x}_1 \\ h_1(\mathbf{x}_2) & h_2(\mathbf{x}_2) & \dots & h_L\mathbf{x}_2 \\ \vdots & \vdots & \ddots & \vdots \\ h_1(\mathbf{x}_N) & h_2(\mathbf{x}_N) & \dots & h_L\mathbf{x}_N \end{bmatrix} \tag{28}$$

$\mathbf{T}$ is the training data matrix

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \mathbf{t}_2^T \\ \vdots \\ \mathbf{t}_N^T \end{bmatrix} \tag{29}$$

and |˙| is the Frobenius norm. The optimal solution is given by $\boldsymbol{\eta}^\star = \mathbf{H}^\dagger T$, where $\mathbf{H}^\dagger$ is the Moore-Penrose generalized inverse of matrix $\mathbf{H}$.

The incremental training version of the ELM method involves solving the mapping function using only blocks of new data. One approach is the online sequential ELM (OS-ELM), (Huang et al. 2015; Nan-Ying Liang et al. 2006). In this approach when i=1 the ELM approach is solved as usual, where

$$\mathbf{H}_1 = \begin{bmatrix} \mathbf{h}_1(\mathbf{x}_{1,1}) \\ \mathbf{h}_2(\mathbf{x}_{1,2}) \\ \vdots \\ \mathbf{h}_L(\mathbf{x}_{1,t}) \end{bmatrix} \tag{30}$$

$$\boldsymbol{\beta}_1 = \mathbf{H}_1^\dagger \mathbf{T}_1 \tag{31}$$

and

$$\mathbf{P}_0 = (\mathbf{H}_1^T \mathbf{H}_1)^{-1} \tag{32}$$

For each i

$$\mathbf{H}_{1:i+1} = \begin{bmatrix} \mathbf{H}_{1:i}^T & \mathbf{H}_i^T \end{bmatrix}^T \tag{33}$$

and then

$$\boldsymbol{\beta}_{1:i+1} = \boldsymbol{\eta}_{1:i} + \mathbf{P}_{1:i+1}\mathbf{H}_{1:i+1}^T(\mathbf{T}_{1:i+1} - \mathbf{H}_{1:i+1}\boldsymbol{\beta}_{1:i}) \tag{34}$$

$$\begin{aligned}\mathbf{P}_{1:i+1} &= \mathbf{P}_{1:i} \\ &- \mathbf{P}_{1:i}\mathbf{H}_{1:i}^T(\mathbf{I} + \mathbf{H}_{1:i+1}\mathbf{P}\mathbf{H}_{1:i+1}^T)^{-1}\mathbf{H}_{1:i+1}\mathbf{P}_{1:i}\end{aligned} \tag{35}$$

SVM

Support vector machines (SVM) are another popular approach to train an optimal predictor. SVM are a two step process, generally. The first step is to project the input vectors/data into a high dimensional space and identify a hyperplane that minimizes the training error. The second step consists of a structural risk minimziatioon calculation, where the optimal hyperplane selected is the (simplest) one that maximizes the margin of seperattion between the bounds of the classes within the training set. This second step differentiates SVM from many other approaches such as decision trees or Naïve Bayes as these are just experience risk methods. The structural risk minimziation step of SVM helps to identify the simplest hyperplane that can function as a predictor, given the data.

Methods to convert a traditional SVM approach to train an optimal predictor in an incremental way try to preserve the support vectors in each increment as they are the most relevant to the margin and risk minimization computation. Then other (older) data samples can be pruned in order to keep the size of the training dataset small. Typically incremental SVM is slow and its speed and performance are dependent on the kernel used to project the data into the space and the kernel used to compute the margins.

Conclusion

Towards the development of an Internet of Battlefield Things (IoBT), capable of leveraging mixed commercial and military IoT technologies, several unique challenges of the tactical environment present themselves. These challenges include development of methods for: (I) quickly gathering training data that reflects unforeseen learning/classification tasks; (II) incrementally learning over real-time data streams; (III) management of limited network bandwidth and connectivity between IoBT assets in data gathering and classification tasks.

In surveying over classical and modern statistical learning theory, this paper has aimed to emphasize how numerical optimization can be used to solve corresponding mathematical problems in these methods. In doing so, this paper aims to encourage the IoT and machine learning research communities to revisit the underlying mathematical underpinnings of stream-based learning, as applicable to IoBT-based systems.

References

Agrawal, R., and Bala, R. 2008. Incremental Bayesian classification for multivariate normal distribution data. *Pattern Recognition Letters* 29(13):1873–1876.

Anderson, J. R., and Matessa, M. 1992. Explorations of an incremental, Bayesian algorithm for categorization. *Machine Learning* 9(4):275–308.

Bartlett, P. L.; Boucheron, S.; and Lugosi, G. 2002. Model selection and error estimation. *Machine Learning* 48(1):85–113.

Bertsimas, D., and Tsitsiklis, J. 1993. Simulated annealing. *Statistical Science* 10–15.

Bottou, L., and Cun, Y. L. 2004. Large scale online learning. In *Advances in neural information processing systems*, 217–224.

Bottou, L. 1998a. Online algorithms and stochastic approximations. In Saad, D., ed., *Online Learning and Neural Networks*. Cambridge, UK: Cambridge University Press.

Bottou, L. 1998b. Online learning and stochastic approximations. *On-line learning in neural networks* 17(9):142.

Boyd, S., and Vanderberghe, L. 2004. *Convex Programming*. New York, NY: Wiley.

Byrd, R. H.; Hansen, S. L.; Nocedal, J.; and Singer, Y. 2016. A stochastic quasi-newton method for large-scale optimization. *SIAM Journal on Optimization* 26(2):1008–1031.

Calandriello, D.; Lazaric, A.; and Valko, M. 2017. Second-order kernel online convex optimization with adaptive sketching. In *International Conference on Machine Learning*.

Castro, R. 2015. 2di70-statistical learning theory lecture notes.

Cauwenberghs, G., and Poggio, T. Incremental and Decremental Support Vector Machine Learning. In *Neural Information Processing Systems (NIPS)*, 388–394. MIT Press Cambridge, MA, USA.

Dai, B.; Xie, B.; He, N.; Liang, Y.; Raj, A.; Balcan, M.-F. F.; and Song, L. 2014. Scalable kernel methods via doubly stochastic gradients. In *Advances in Neural Information Processing Systems*, 3041–3049.

Defazio, A.; Bach, F.; and Lacoste-Julien, S. 2014. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems*, 1646–1654.

Dekel, O.; Shalev-Shwartz, S.; and Singer, Y. 2006. The forgetron: A kernel-based perceptron on a fixed budget. In *Advances in Neural Information Processing Systems 18*, 259266. MIT Press.

Diehl, C., and Cauwenberghs, G. 2003. Svm incremental learning, adaptation and optimization. *Proceedings of the International Joint Conference on Neural Networks, 2003.* 4(x):2685–2690.

Elad, M., and Aharon, M. 2006. Image denoising via sparse and redundant representations over learned dictionaries. *Trans. Img. Proc.* 15(12):3736–3745.

Engel, Y.; Mannor, S.; and Meir, R. 2004. The kernel recursive least-squares algorithm. *IEEE Transactions on Signal Processing* 52(8):2275–2285.

Facchinei, F.; Scutari, G.; and Sagratella, S. 2015. Parallel selective algorithms for nonconvex big data optimization. *IEEE Transactions on Signal Processing* 63(7):1874–1889.

Goldfarb, D. 1970. A family of variable metric updates derived by variational means. *Mathematics of Computation* 24(109):23–26.

Graves, A.; Mohamed, A.-r.; and Hinton, G. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*, 6645–6649. IEEE.

Hastie, T.; Tibshirani, R.; and Friedman, J. 2009. Overview of supervised learning. In *The elements of statistical learning*. Springer. 9–41.

Haykin, S. 1994. Neural networks: A comprehensive foundation.

Hu, C.; Pan, W.; and Kwok, J. T. 2009. Accelerated gradient methods for stochastic optimization and online learning. In *Advances in Neural Information Processing Systems*, 781–789.

Huang, G.; Huang, G.-B.; Song, S.; and You, K. 2015. Trends in extreme learning machines: A review. *Neural Networks* 61:32–48.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2016. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision* 116(1):1–20.

Johnson, R., and Zhang, T. 2013. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems*, 315–323.

Jolliffe, I. 1986. *Principal Component Analysis*. Springer Verlag.

Jurafsky, D., and Martin, J. 2017. Hidden Markov Models. *Speech and Language Processing* (Chapter 20):21.

Kimeldorf, G., and Wahba, G. 1971. Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications* 33(1):82–95.

Kivinen, J.; Smola, A. J.; and Williamson, R. C. 2004. Online Learning with Kernels. *IEEE Transactions on Signal Processing* 52:2165–2176.

Koltchinskii, V., et al. 2009. Sparsity in penalized empirical risk minimization. In *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, volume 45, 7–57. Institut Henri Poincaré.

Koppel, A.; Fink, J.; Warnell, G.; Stump, E.; and Ribeiro, A. 2016a. Online learning for characterizing unknown environments in ground robotic vehicle models. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, 626–633. IEEE.

Koppel, A.; Warnell, G.; Stump, E.; and Ribeiro, A. 2016b. Parsimonious online kernel learning via sparse projections in function space. *Journal of Machine Learning Research (submitted)*.

Kott, A.; Swami, A.; and West, B. J. 2016. The internet of battle things. *Computer* 49(12):70–75.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Le, T.; Nguyen, T.; Nguyen, V.; and Phung, D. 2016a. Dual space gradient descent for online learning. In *Advances in Neural Information Processing Systems*, 4583–4591.

Le, T.; Nguyen, V.; Nguyen, T. D.; and Phung, D. 2016b. Nonparametric budgeted stochastic gradient descent. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, 654–662.

Lee, J. D.; Simchowitz, M.; Jordan, M. I.; and Recht, B. 2016. Gradient descent only converges to minimizers. In *Conference on Learning Theory*, 1246–1257.

Lian, X.; Huang, Y.; Li, Y.; and Liu, J. 2015. Asynchronous parallel stochastic gradient for nonconvex optimization. In *Advances in Neural Information Processing Systems*, 2737–2745.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Liu, W.; Pokharel, P. P.; and Principe, J. C. 2008. The kernel least-mean-square algorithm. *Signal Processing, IEEE Transactions on* 56(2):543–554.

Losing, V.; Hammer, B.; and Wersing, H. 2018. Incremental on-line learning: A review and comparison of state of the art algorithms. *Neurocomputing* 275:1261–1274.

Lu, J.; Hoi, S. C.; Wang, J.; Zhao, P.; and Liu, Z.-Y. 2016. Large scale online kernel learning. *The Journal of Machine Learning Research* 17(1):1613–1655.

Mairal, J.; Bach, F.; and Ponce, J. 2012. Task-driven dictionary learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34(4):791–804.

Mairal, J.; Bach, F.; Ponce, J.; Sapiro, G.; and Zisserman, A. 2008. Supervised dictionary learning. In *Advances in Neural Information Processing Systems 21, Proceedings of the Twenty-Second Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 8-11, 2008*, 1033–1040.

Mairal, J.; Koniusz, P.; Harchaoui, Z.; and Schmid, C. 2014. Convolutional kernel networks. In *Advances in Neural Information Processing Systems*, 2627–2635.

Mairal, J.; Elad, M.; and Sapiro, G. 2007. Sparse representation for color image restoration. In *the IEEE Trans. on Image Processing*, 53–69. ITIP.

Mairal, J. 2016. End-to-end kernel learning with supervised convolutional kernel networks. In *Advances in Neural Information Processing Systems*, 1399–1407.

Mallat, S. 2008. *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, 3rd edition.

Metsis, V.; Androutsopoulos, I.; and Paliouras, G. 2006. Spam filtering with naive bayes-which naive bayes? In *Ceas*, 9.

Mokhtari, A., and Ribeiro, A. 2015. Global convergence of online limited memory bfgs. *Journal of Machine Learning Research* 16:3151–3181.

Mokhtari, A.; Koppel, A.; Scutari, G.; and Ribeiro, A. 2017. Large-scale nonconvex stochastic optimization by doubly stochastic successive convex approximation. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 4701–4705. IEEE.

Mokhtari, A.; Gürbüzbalaban, M.; and Ribeiro, A. 2016. Surpassing gradient descent provably: A cyclic incremental method with linear convergence rate. *arXiv preprint arXiv:1611.00347*.

Murphy, K. 2012. *Machine Learning: A Probabilistic Perspective*. MIT press.

Murty, K. G., and Kabadi, S. N. 1987. Some np-complete problems in quadratic and nonlinear programming. *Mathematical programming* 39(2):117–129.

Nan-Ying Liang; Guang-Bin Huang; Saratchandran, P.; and Sundararajan, N. 2006. A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. *IEEE Transactions on Neural Networks* 17(6):1411–1423.

Nelder, J. A., and Baker, R. J. 1972. Generalized linear models. *Encyclopedia of statistical sciences*.

Nemirovski, A.; Juditsky, A.; Lan, G.; and Shapiro, A. 2009. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization* 19(4):1574–1609.

Norkin, V., and Keyzer, M. 2009. On stochastic optimization and statistical learning in reproducing kernel hilbert spaces by support vector machines (svm). *Informatica* 20(2):273–292.

Paternain, S.; Mokhtari, A.; and Ribeiro, A. 2017. A second order method for nonconvex optimization. *arXiv preprint arXiv:1707.08028*.

Pati, Y.; Rezaiifar, R.; and Krishnaprasad, P. 1993. Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition. In *Proceedings of the Asilomar Conference on Signals, Systems and Computers*.

Raginsky, M.; Rakhlin, A.; and Telgarsky, M. 2017. Nonconvex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. *arXiv preprint arXiv:1702.03849*.

Richard, C.; Bermudez, J. C. M.; and Honeine, P. 2009. Online prediction of time series data with kernels. *IEEE Transactions on Signal Processing* 57(3):1058–1067.

Robbins, H., and Monro, S. 1951. A stochastic approximation method. *Ann. Math. Statist.* 22(3):400–407.

Ruping, S. 2001. Incremental Learning with Support Vector Machines. *Learning* 0–1.

Scardapane, S., and Di Lorenzo, P. 2017. Stochastic training of neural networks via successive convex approximations. *arXiv preprint arXiv:1706.04769*.

Schmidt, M.; Roux, N. L.; and Bach, F. 2013. Minimizing finite sums with the stochastic average gradient. *arXiv preprint arXiv:1309.2388*.

Schölkopf, B.; Herbrich, R.; and Smola, A. J. 2001. A generalized representer theorem. *Subseries of Lecture Notes in Computer Science Edited by JG Carbonell and J. Siekmann* 416.

Shanno, D. F., and Phua, K. H. 1976. Algorithm 500: Minimization of unconstrained multivariate functions [e4]. *ACM Transactions on Mathematical Software (TOMS)* 2(1):87–94.

Shawe-Taylor, J.; Bartlett, P. L.; Williamson, R. C.; and Anthony, M. 1998. Structural risk minimization over data-dependent hierarchies. *IEEE transactions on Information Theory* 44(5):1926–1940.

Suri, N.; Tortonesi, M.; Michaelis, J.; Budulas, P.; Benincasa, G.; Russell, S.; Stefanelli, C.; and Winkler, R. 2016. Analyzing the applicability of internet of things to the battlefield environment. In *2016 International Conference on Military Communications and Information Systems (ICMCIS)*, 1–8.

Vapnik, V. N. 1995. *The Nature of Statistical Learning Theory*. New York: Springer-Verlag.

Vapnik, V. 2013. *The nature of statistical learning theory*. Springer science & business media.

Wheeden, R.; Wheeden, R.; and Zygmund, A. 1977. *Measure and Integral: An Introduction to Real Analysis*. Chapman & Hall/CRC Pure and Applied Mathematics. Taylor & Francis.

Zang, W.; Zhang, P.; Zhou, C.; and Guo, L. 2014. Comparative study between incremental and ensemble learning on data streams: Case study. *Journal Of Big Data* 1(1):5.

Zheng, D. E., and Carter, W. A. 2015. Leveraging the internet of things for a more efficient and effective military.

Zhu, J., and Hastie, T. 2005. Kernel Logistic Regression and the Import Vector Machine. *Journal of Computational and Graphical Statistics* 14(1):185–205.

Challenges and Characteristics of Intelligent Autonomy for Internet of Battle Things in Highly Adversarial Environments

Alexander Kott

U.S. Army Research Laboratory, Adelphi, MD, USA
alexander.kott1.civ@mail.mil

Abstract

Numerous, artificially intelligent, networked things will populate the battlefield of the future, operating in close collaboration with human warfighters, and fighting as teams in highly adversarial environments. This paper explores the characteristics, capabilities and intelligence required of such a network of intelligent things and humans – Internet of Battle Things (IOBT). It will experience unique challenges that are not yet well addressed by the current generation of AI and machine learning.

Introduction

Internet of Intelligent Battle Things is the emerging reality of warfare. A variety of networked intelligent systems – things – will continue to proliferate on the battlefield, where they will operate with varying degrees of autonomy. Intelligent things will not be a rarity but a ubiquitous presence on the future battlefield. (Scharre 2014)

Most of such intelligent things will not be too dissimilar from the systems we see on today's battlefield, such as unattended ground sensors, guided missiles (especially the fire-and-forget variety) and of course the unmanned aerial systems (UAVs). They will likely include physical robots ranging from very small size (such as an insect-scale mobile sensors) to large vehicle that can carry troops and supplies. Some will fly, others will crawl or walk or ride. Their functions will be diverse. Sensing (seeing, listening, etc.) the battlefield will be one common function. Numerous small, autonomous sensors can cover the battlefield and provide an overall awareness to the warfighters that is reasonably compete and persistent (Fig. 1).

Other things might acts as defensive devices, e.g., autonomous active protection systems (Freedberg 2016). Finally, there will be munitions that are intended to impose physical or cyber effects on the enemy. These will not be autonomous. Instead, they will be controlled by human warfighters. This assumes that the combatants of that future battlefield will comply with a ban on offensive autonomous weapons beyond meaningful human control. Although the US Department of Defense already imposes strong restrictions on autonomous and semi-autonomous weapon systems (Hall 2017), nobody can predict what other countries might decide on this matter.

In addition to physical intelligent things, the battlefield – or at least the cyber domain of the battlefield -- will be populated with disembodied, cyber robots. These will reside within various computers and networks, and will move and acts in the cyberspace. Just like physical robots, the cyber robots will be employed in a wide range of roles. Some will protect communications and information (Stytz et al. 2005) or will fact-check, filter and fuse information for cyber situational awareness (Kott et al. 2014). Others will defend electronic devices from effects of electronic warfare. These defensive actions might include creation of informational or electromagnetic deceptions or camouflage. Yet others will act as situation analysts and decision advisers to the humans or physical robots. In addition to these defensive or advisory roles, cyber robots might also take on more assertive functions, such as executing cyber actions against the enemy systems (Fig. 2).

In order to be effective in performing these functions, battle things will have to collaborate with each other, and also with the human warfighters. This will require a significant degree of autonomous self-organization; and also of accepting a variety of relations between things and humans, e.g., from complete autonomy of an unattended ground sensor to a tight control of certain systems, and these modes will have to change flexibly as needed. Priorities, objectives, and rules of engagement will change rapidly, and intelligent things will have to adjust accordingly (Kott et al. 2016).

Clearly, these requirements imply a high degree of intelligence on part of the things. Particularly important is the necessity to operate in a highly adversarial environment,

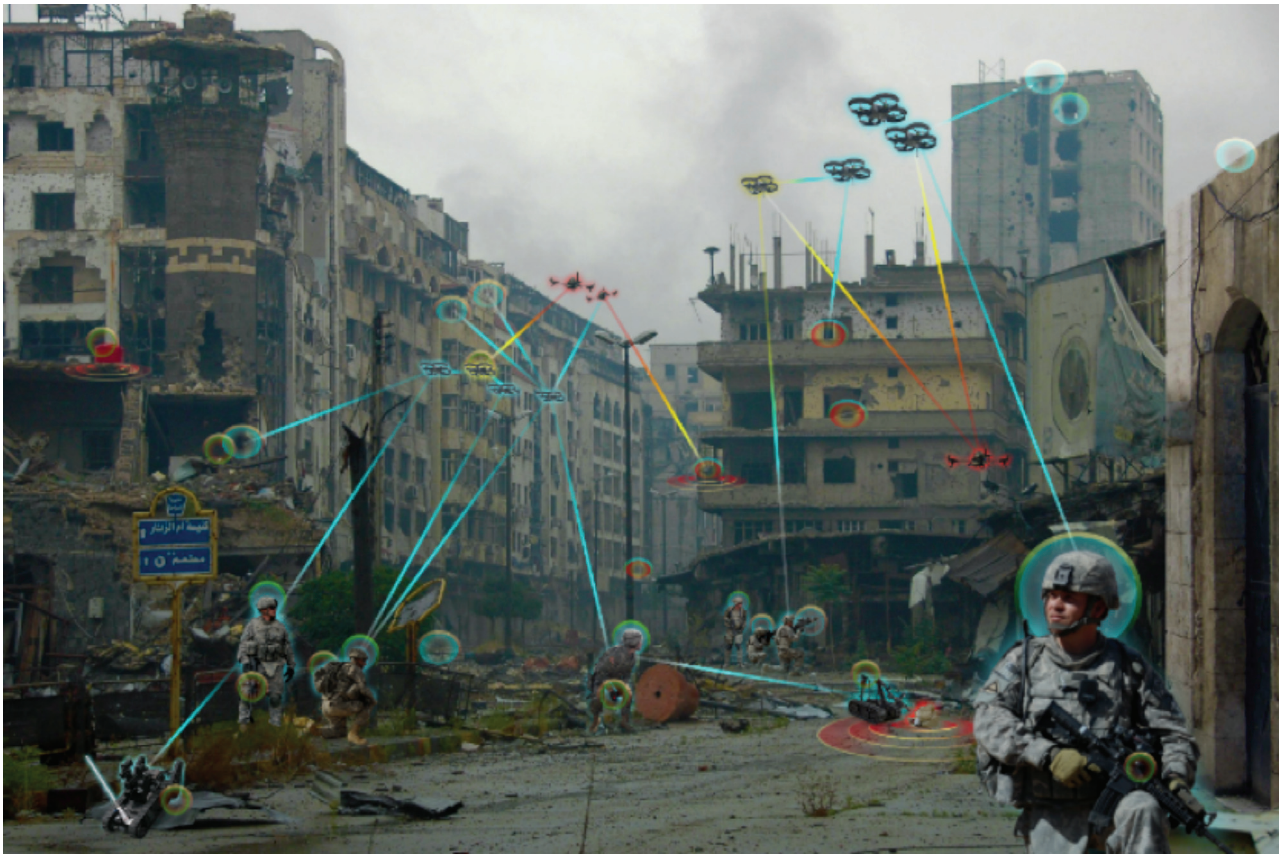


Figure 1 Networked teams of intelligent things and humans will operate in extremely complex, challenging environment: unstructured, unstable, rapidly changing, chaotic, rubble-filled, adversarial and deceptive.

i.e., intentionally hostile and not merely randomly dangerous world. The intelligent things will have to constantly think about an intelligent adversary that strategizes to deceive and defeat them. Without this adversarial intelligence, the battle things will not survive long enough to be useful.

The Challenges of Autonomous Intelligence on the Battlefield

The vision – or rather the emerging reality -- of the battlefield populated by intelligent things, portends a multitude of profound challenges. While use of AI for battlefield tasks has been explored on multiple occasions, e.g., (Rasch et al. 2002), and AI makes things individually and collectively more intelligent, it also makes the battlefield harder to understand and to manage. Human warfighters have to face a much more complex, more unpredictable world where things have the mind of their own and perform actions that may appear inexplicable to the humans. Direct control of such intelligent things becomes impossible or limited to cases of decisions whether to take a specific destructive action.

On the other hand, humans complicate the life for intelligent things. Human and things think differently. Intelligent things, in the foreseeable future, will be challenged in understanding and anticipating human intent, goals, lines of reasoning and decisions. Humans and things will remain largely opaque to each other. And yet, things will be expected to perceive, reason and act while taking into account the social, cognitive and physical needs of their human teammates. Furthermore, things will often deal with humans who are experiencing extreme physical and cognitive stress, and therefore may behave differently from what can be assumed from observing humans under more benign conditions

An intelligent thing will need to deal with a world of astonishing complexity. The sheer number and diversity of things – and humans – within the IoBT will be enormous. The number of connected things, for example within a future Army brigade, is likely to be several orders of magnitude greater than in current practice. This, however, is just the beginning. Consider that intelligent things belonging to

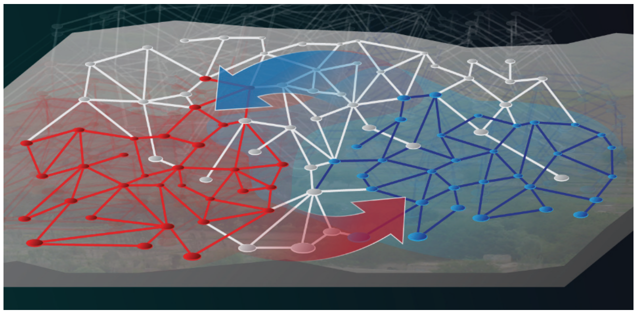


Figure 2 Networks of the opponents will fight each other with cyber and electromagnetic attacks of great diversity and volume; most of such offensive and defensive actions will be performed by autonomous cyber agents.

such a brigade will inevitably interact – willingly or unwillingly -- with things owned and operated by other parties, such as those of the adversary or owned by the surrounding civilian population. If the brigade operates in a large city, where each apartment building can contains thousands of things, the overall universe of connected things grows to enormous numbers. Million things per square kilometer is not an unreasonable expectation (Fig. 2).

The above scenario also points to a great diversity of things within the overall environment of the battlefield. Things will come from different manufacturers, with different designs, capabilities, and purposes, configured or machine-learned differently, etc. No individual thing will be able to use pre-conceived (pre-programmed, pre-learned, etc.) assumptions about behaviors or performance of other things it meets on the battlefield. Instead, behaviors and characteristics will have to be learned and updated autonomously dynamically during the operations. That includes humans – yes, humans are a specie of things, in a way – and therefore the behaviors and intents of humans, such as friendly warfighters, adversaries, and civilians and so on, will have to be continually learned and inferred.

The cognitive processes of both things and humans will be severely challenged in this environment of voluminous and heterogeneous information. Rather than the communications bandwidth, the cognitive bandwidth may become the most severe constraint. Both humans and things seek information that is well-formed, reasonably sized, essential in nature, and highly relevant to their current situation and mission. Unless information is useful, it does more harm than good. The trustworthiness of the information and the value of information arriving from different sources, especially other things, will be highly variable and generally uncertain. For any given intelligent thing, the incoming information could contain mistakes, erroneous observations or conclusions made by other things, or intentional distortions – deceptive information – produced by an adversary malware residing on friendly things or otherwise inserted into the environment. Both humans and things are susceptible to deception, and humans are likely to experience cognitive challenges when surrounded by opaque things that might be feeding the humans with untrustworthy information (Kott and Alberts 2017).

This reminds us that the adversarial nature of the battlefield environment is a concern of exceptional importance, above all others. The intelligent things will have to deal with an intelligent, capable adversary. The adversary will apply to things physical destruction, either by means such as gunfire, also known as “kinetic” effects, or by using directed energy weapons. The adversary will be jamming the channels of communications between things, and between things and humans. The adversary will deceive things by presenting them with misleading information. Recent research in adversarial learning comes to mind in this connection (Papernot et al. 2016). Perhaps most dangerously, the adversary will attack intelligent things by depositing malware on them.

AI will Fight the Cyber Adversary

A key assumption that must be taken regarding the IoBT is that in a conflict with a technically sophisticated adversary, IoBT will be a heavily contested battlefield (Kott 2015). Enemy software cyber agents -- malware -- will infiltrate our network and attack our intelligent things. To fight them, things will need artificial cyber hunters - intelligent, autonomous, mobile agents specialized in active cyber defense and residing on IoBT.

Such agents will stealthily patrol the networks, detect the enemy malware while remaining concealed, and then destroy or degrade the enemy malware. They will do so mostly autonomously, because human cyber experts will be always scarce on the battlefield. They will be adaptive because the enemy malware is constantly evolving. They will be stealthy because the enemy malware will try to find and kill them. At this time, such capabilities do not exist but are a topic of research (Theron et al. 2018)). Here, let’s explore the desired characteristics of an intelligent autonomous agent operating in the context of IoBT.

We consider a thing – a simple senor or a complex military vehicle -- with one or more computers residing on the thing. Each computer contributes considerably into operation of the thing or systems installed on the thing. One or more of the computers are assumed to have been compromised, where the compromise is either established as a fact, or is suspected.

Due to the contested nature of the communications environment (e.g., the enemy is jamming the communications or radio silence is required to avoid detection by the enemy), communications between the thing and other elements of the friendly force can be limited and intermittent. Under some conditions, communications are entirely impossible.

Given the constraints on communications, conventional centralized cyber defense is infeasible. (Here centralized cyber defense refers to an architecture where local sensors send cyber-relevant information to a central location, where highly capable cyber defense systems and human analysts detect the presence of malware and initiate corrective actions remotely). It is also unrealistic to expect that the human war-fighters in the vicinity of the thing (if such exist) have the necessary skills or time available to perform cyber defense functions for that thing.

Therefore, cyber defense of the thing and its computing devices has to be performed by an intelligent, autonomous software agent. The agent (or multiple agents per thing) would stealthily patrol the networks, detect the enemy agents while remaining concealed, and then destroy or degrade the enemy malware. The agent has to do so mostly autonomously, without support or guidance of a human expert.

In order to fight the enemy malware deployed on the friendly thing, the agent often has to take destructive actions, such as deleting or quarantining certain software. Such destructive actions are carefully controlled by the appropriate rules of engagement, and are allowed only on the computer where the agent resides. The agent may also be the primary mechanism responsible for defensive cyber maneuvering (of which mobbing target defense is an example), deception, e.g., redirection of malware to honeypots (De Gaspari et al. 2016), self-healing, e.g., (Azim et al. 2014), and other such autonomous or semi-autonomous behaviors (Jajodia et al. 2011).

The actions of the agent, in general, cannot be guaranteed to preserve availability of integrity of the functions and data of friendly computers. There is a risk that an action of the agent may "break" the friendly computer, disable important friendly software, or corrupt or delete important data. This risk, in a military environment, has to be balanced against the death or destruction caused by the enemy if the agent's action is not taken.

Provisions are made to enable a remote or local human controller to fully observe, direct and modify the actions of the agent. However, it is recognized that human control is often impossible, especially because of intermittent communications. The agent, therefore, is able to plan, analyze and perform most or all of its actions autonomously. Similarly, provisions are made for the agent to collaborate with other agents (who reside on other computers); however, in most cases, because the communications are impaired or observed by the enemy, the agent operates alone.

The enemy malware, and its capabilities and techniques, evolve rapidly. So does the environment in general, together with the mission and constraints that the thing is subject to. Therefore, the agent is capable of autonomous learning.

Because the enemy malware knows that the agent exists and is likely to be present on the computer, the enemy malware seeks to find and destroy the agent. Therefore, the agent possesses techniques and mechanisms for maintaining a degree of stealth, camouflage and concealment. More generally, the agent takes measures that reduce the probability that the enemy malware may detect the agent. The agent is mindful of the need to exercise self-preservation and self-defense.

It is assumed here that the agent resides on a computer where it was originally installed by a human controller or by an authorized process. We do envision a possibility that an agent may move itself (or move a replica of itself) to another computer. However, such propagation is assumed to occur only under exceptional and well-specified conditions, and to take place only within friendly network – from one friendly computer to another friendly computer. This brings to mind the controversy about "good viruses." Such viruses have been proposed, criticized and dismissed earlier (Muttik 2016). These criticisms do not apply here. This agent is not a virus because it does not propagate except under explicit conditions within authorized and cooperative nodes. It is also used only in military environments where most usual concerns do not apply.

AI will have to Advance Significantly

Agents will have to become useful team-mates – not tools -- of human warfighters on a highly complex and dynamic battlefield. Consider Fig. 1 that depicts an environment in which a highly-dispersed team of human and intelligent agents (including but not limited to physical robots) is attempting to access a multitude of highly heterogeneous and uncertain information sources, and use them for forming situational awareness and making decision (Kott et al. 2011), all while trying to survive extreme physical and cyber threats. They must be effective, in this unstructured, unstable, rapidly changing, chaotic, rubble-filled adversarial environments; learning in real-time, under extreme time constraints, with only a few observations that are potentially erroneous, of uncertain accuracy and meaning, or even intentionally misleading and deceptive (Fig. 3).

Clearly, it is far beyond the current state of AI to operate intelligently in such an environments and with such demands. In particular, Machine Learning – an area that has seen a dramatic progress in the last decade – must experience major advances in order to become relevant to the real battlefield. Let's review some of the required advances.

Learning with very small number of samples is clearly a necessity in an environment where the enemy and friends change the tactics continuously, and the environment itself

Figure 3 . AI-enabled agents -- members of a human-agent team – will rapidly learn in ever-changing, complex environments, providing the team's commander with real-time estimates of enemy, reasoning on possible courses of action, and tactically sensible decision.

is highly fluid, rich with details, dynamic and changing rapidly. Furthermore, very few if any of the available samples will be labelled, or at least not in a very helpful manner.

A typical sample might be a video snippet of events and physical surroundings or a robot, for example, where the overwhelming majority of elements (e.g., pieces of rubble) are hardly relevant and potentially misleading for the purposes of learning. The information of the samples is likely to be highly heterogeneous of nature. Depending on circumstances, samples might consist of one or more of the following: still images in various parts of the spectrum (IR, visible, etc.); video; audio; telemetry data; solid models of the environment; records of communications between agents; and so on.

Some samples may be misleading in general, even if unintentionally (e.g., an action succeeds even though an unsuitable action is applied) and the machine learning algorithms will have to make the distinction between relevant and irrelevant, instructive and misleading. In addition, some of the samples might be a product of intentional deception by the enemy. In general, issues of Adversarial Learning and Adversarial Reasoning are of great importance (Papernot et al. 2016).

Yet another challenge that is uniquely exacerbated by battlefield conditions are constraints on the available electric power. Most successful AI relies on vast computing and electrical power resources including cloud-computing reach-back when necessary. The battlefield AI, on the other hand, must operate within the constraints of edge devices, such as small sensors, micro-robots, and handheld radios of warfighters. This means that computer processors must be relatively lights and small, and as frugal as possible in the use of electrical power. One might suggest that a way to overcome such limitations on computing resources available directly on the battlefield is to offload the computations via wireless communications to a powerful computing resource located outside of the battlefield. Unfortunately, it is a viable solution, because the enemy's inevitable interference with friendly networks will limit the opportunities for use of reach-back computational resources.

A team that includes multiple warfighters and multiple artificial agents must be capable of distributed learning and reasoning. Besides distributed learning, these include such challenges as: multiple decentralized mission-level task allocation; self-organization, adaptation, and collaboration; space management operations; and joint sensing and perception. Commercial efforts to date have been largely limited to single platforms in benign settings. Military-focused programs like the MAST CTA (Piekarski et al. 2017), have been developing collaborative behaviors for UAVs. Ground vehicle collaboration is challenging and is largely still at the basic research level at present. In particular, to address such challenges, a new collaborative research alliance called Distributed and Collaborative Intelligent Systems and Technology (DCIST) has been initiated (https://dcist-cra.org/). Note that the battlefield environment imposes yet another complication: because the enemy interferes with communications, all this collaborative, distributed AI must work well even with limited, intermittent connectivity.

Humans in the Ocean of Things

In this vision of the future warfare, a key challenge is to enable autonomous systems and intelligent agents to effectively and naturally interact across a broad range of warfighting functions. Human-agent collaboration is an active ongoing research area. It must address such issues as trust and transparency, common understanding of shared perceptions, and human-agent dialog and collaboration.

One seemingly relevant technology is Question Answering—the system's ability to respond with a relevant, correct information to a clearly stated question. Successes of commercial technologies of question-answering are indisputable. They work well for very large, stable, and fairly accurate volumes of data, e.g., encyclopedias. But such tools don't work for rapidly changing battlefield data, also distorted by adversary's concealment and deception. They cannot support continuous, meaningful dialog in which both warfighters and artificially intelligent agents develop shared situational awareness and intent understanding. Research is being performed to develop human-robotic dialog technology for warfighting tasks, using natural voice, which is critical for reliable battlefield teaming.

A possible approach to developing the necessary capabilities – both human and AI – is to train a human-agent team in immersive artificial environments. This requires building realistic, intelligent entities in immersive simulations. Training (for humans) and learning (for agents) experiences must exhibit high degree of realism to match operational demands. Immersive simulations for human training and machine learning must have physical and sociocultural interactions with high fidelity and realistic complexity of the operational environment. These include realistic behaviors of human actors (friendly warfighters, enemies, non-combatants), and interactions and teaming with robots and other intelligent agents. In today's video games, these interactions are limited and not suitable for simulating real battlefield. Advances in AI are needed to drive the character behaviors that are truly realistic, diverse, and intelligent.

To this end, some of the cutting-edge efforts in computer-generation of realistic virtual characters are moving towards what would be needed to enable realistic interactions in an artificial immersive battlefield. For example, Hollywood studios on a number of occasions sought technologies of the Army-sponsored Institute for Creative Technologies (http://ict.usc.edu/)to create realistic avatars of actors. These technologies enable film creators to digitally insert an actor into scenes, even if that actor is unavailable, much older or younger, or deceased. That's how actor Paul Walker was able to appear in "Fast and Furious 7," even though he died partway into filming (CBS News 2017).

Summary

Intelligent things – networked and teamed with human warfighters – will be a ubiquitous presence on the future battlefield. Their appearances, roles and functions will be highly diverse. The artificial intelligence required for such things will have to be significantly greater than what is provided by today's AI and machine learning technologies. Adversarial – strategically and not randomly dangerous -- nature of the battlefield is a key driver of these requirements. Complexity of the battlefield – including the complexity of collaboration with humans – is another major driver. Cyber warfare will assume a far greater importance, and it will be AI that will have to fight cyber adversaries. Major advances in areas such as adversarial learning and adversarial reasoning will be required. Simulated immersive environments may help to train the humans and to train AI.

References

Azim, M.T.; Neamtiu, I; and Marvel, L.M. 2014. Towards self-healing smartphone software via automated patching. In *Proceedings of the 29th ACM/IEEE international conference on Automated software engineering*, 623-628. ACM.

CBS News. 2017. Digital doubles: Bringing actors back to life, February 26, 2017, online at https://www.cbsnews.com/news/digital-doubles-bringing-actors-back-to-life/

De Gaspari, F.; Jajodia, S.; Mancini, L.V.; and Panico, A. 2016, October. AHEAD: A New Architecture for Active Defense. In *Proceedings of the 2016 ACM Workshop on Automated Decision Making for Active Cyber Defense*, 11-16. Vienna, Austria: ACM.

Freedberg, S. J. Jr. March 09, 2016. Missile Defense for Tanks: Raytheon Quick Kill vs. Israeli Trophy, Breakingdefense.com

Jajodia, S.; Ghosh, A.K.; Swarup, V.; and Wang, X.S. eds. 2011. *Moving target defense: creating asymmetric uncertainty for cyber threats (Vol. 54).* Springer Science & Business Media.

Hall, B. K. July 2017. Autonomous Weapons Systems Safety, *Joint Force Quarterly* 86, 86-93, online at http://ndupress.ndu.edu/JFQ/Joint-Force-Quarterly-86/Article/1223911/autonomous-weapons-systems-safety/

Kott, A.; Wang, C.; and Erbacher, R. F., eds. 2014. *Cyber Defense and Situational Awareness.* New York: Springer.

Kott, A.; Alberts, D.S.; and Wang, C. 2015. Will Cybersecurity Dictate the Outcome of Future Wars? *Computer*, *48*(12): 98-101.

Kott, A.; Singh, R.; McEneaney, W. M.; and Milks, W. 2011. Hypothesis-driven information fusion in adversarial, deceptive environments. *Information Fusion*, 12(2): 131-144.

Kott, A.; and Alberts, D. S. How Do You Command an Army of Intelligent Things? 2017. *Computer* 12: 96-100

Kott, A.; Swami, A.; and West, B. J. 2016. The Internet of Battle Things. *Computer* 49, no. 12: 70-75.

Muttik, I. 2016, Good Viruses. Evaluating the Risks, Talk at DEFCON-2016 Conference, online at https://www.defcon.org/images/defcon-16/dc16-presentations/defcon-16-muttik.pdf

Papernot, N.; McDaniel, P.; Jha, S.; Fredrikson, M.; Celik, Z.B.; and Swami, A. 2016, March. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium,* 372-387. IEEE.

Piekarski, B.; Mathis, A.; Nothwang, W.; Baran, D.; Kroninger, C.; Sadler, B.; Matthies, L.; Kumar, V.; Chopra, I.; Humbert, S.; and Sarabandi, K. 2017. *Micro Autonomous Systems and Technology (MAST) 2016 Annual Report for Program Capstone.* Technical Report ARL-SR-0377. US Army Research Laboratory, Adelphi, MD, United States.

Rasch, R.; Kott, A.; and Forbus, K. D. 2002. AI on the battlefield: An experimental exploration. In *Proceedings of the Fourteenth Innovative Applications of Artificial Intelligence Conference on Artificial Intelligence*, Edmonton, Alberta, Canada.

Scharre, P. 2014. Robotics on the Battlefield Part II: The Coming Swarm, Report, Center for a New American Security, Washington, DC.

Stytz, M. R.; Lichtblau, D. E.; and Banks, S. B. 2005. Toward using intelligent agents to detect, assess, and counter cyberattacks in a network-centric environment. Report, Institute for Defense Analyses, Alexandria, VA.

Theron, P.; Kott, A.; Drasar, M.; LeBlanc, B.; Rzadca, K.; Pihelgas, M.; Mancini, L.; and Panico, A. 2018. Towards an active, autonomous and intelligent cyber defense of military systems. *In Proceedings of the ICMCIS-2018 Conference*, Warsaw, Poland, to appear.

Artificial Intelligence for the Internet of Everything

W. F. Lawless,[1] Ranjeev Mittu,[2] Donald Sofge[2]

[1] Paine College, Augusta, GA 30901; [2] Naval Research Laboratory, Washington, DC 20375
[1] w.lawless@icloud.com; [2] {ranjeev.mittu, donald.sofge} @nrl.navy.mil

Abstract

For the Internet of Everything (IoE), from an AI perspective, we discuss the meaning, value and effect that the internet of things (IoT) is expected to have on ordinary life, in industry (IIoT), on the battlefield (IoBT), in the medical field (IoMT) and with intelligent-agent feedback in the form of constructive and destructive interference (IoIT). We consider the topic open-ended but with an AI perspective that addresses how the IoE affects sensing, perception, cognition and behavior, or causal relations whether the context is clear or uncertain for mundane decisions, complex decisions on the battlefield, life and death decisions in the medical arena, or decisions affected by intelligent agents and machines. We pay attention to theoretical perspectives for how these "things" may affect individuals, teams and society; and in turn how they may affect these "things". We are most interested in what may happen when these "things" begin to think. Our ultimate goal is to use AI to advance autonomy and autonomous characteristics to improve the performance of individual agents and hybrid teams of humans, machines, and robots for the betterment of society.

IoE: IoT, IoBT, and IoIT--Background and overview

The Internet of Everything (IoE) [1] generalizes machine-to-machine (M2M) communications for the Internet of Things (IoT) to a more complex system that also encompasses people, robots and machines. From Chambers (2014), IoE connects

> *people, data, process and things. It is revolutionizing the way we do business, transforming communication, job creation, education and healthcare across the globe. ... by 2020, more than 5 billion people will be connected, not to mention 50 billion things. ... [With IoE] [p]eople get better access to education, healthcare and other opportunities to improve their lives and have better experiences. Governments can improve how they serve their citizens and businesses can use the information they get from all these new connections to make better decisions, be more productive and innovate faster.*

This IoT is expected to become big business with a large impact on day-to-day life. From Marr (2015),

> *By 2020, a quarter of a billion vehicles will be connected to the Internet ... new possibilities for in-vehicle services and automated driving. In fact, we already have cars that can drive on their own – Google's self-driving cars currently average about 10,000 autonomous miles per week. ... Machine-to-machine (M2M) connections will grow ... to 27 billion by 2024, with China taking a 21% share and the U.S. 20%. ... the IoT will have a total economic impact of up to $11 trillion by 2025*

The industrial internet of things (IIoT)[2] is impacting industry by forcing alliances to keep pace with innovation; e.g., (Ramachandran et al., 2017):

> *Amazon could help finance a network Dish is building focused on the "Internet of Things"—the idea that everything from bikes to Amazon's drones can have web connectivity everywhere.*

The internet of things (IoT) is "all about connecting objects to the network and enabling them to collect and share data" (Munro, 2017). But a big question is (Alessi, 2017):

> *who will create and dominate a realm of technology ... to become the backbone of industrial automation and provide mountains of data about everything from parts inventories to how products are wearing long after their purchase.*

[1] http://ioeassessment.cisco.com

[2] http://www3.weforum.org/docs/WEFUSA_IndustrialInternet_Report2015.pdf

With the approach of IoT in everyday life (Gasior & Yang, 2013), in industry (IIoT),[3] on battlefields (IoBT),[4] in the medical arena (IoMT),[5] distributed with sensory networks and cyber-physical systems, and even with device-level intelligence (IoIT),[6] some of the known issues identified by Moskowitz (2017) are the explosion of data (e.g., cross-compatible systems; storage locations); security challenges (e.g. adversarial resilience,[7] data exfiltration, covert channels; enterprise protection; privacy); self-* [8] and autonomic behaviors, and the competitive risks to users, teams, enterprises and institutions. Still, despite the pace of rapid advance, "Humans will often be the integral parts of the IoT system" (Stankovic, 2014, p. 4).

For the Internet of Everything, IoT, IoBT, IoMT, IoIT and on will manifest as heterogeneous and potentially self-organizing complex-systems that define human processes, requiring interoperability, just-in-time (JIT) human interactions, and the orchestration of local-adaptation functionalities to achieve human objectives and goals (Suri et. al, 2016).

For military matters, IoBT is about the relation of persons-to-machines; e.g., (Cartwright, 2015):

> *how many men it takes to run a machine to how many machines a man can control.*

There are practical considerations: Whatever the systems used for the benefits afforded, each must be robust to interruptions, to failure, and resilient to every possible perturbation from wear and tear in daily use. For system-wide failures, a system must have manual control backups; user-friendly methods for joining and leaving networks; autonomous updates and backups; and autonomous hardware updates (e.g., similar to re-ordering inventory or goods automatically by a large retailer like Amazon or Wal-Mart). A system must also provide forensic evidence in the event of a mishap, not only with an onboard backup, but also with an automatic backup to the cloud.

IoT is causing creative disruption to commerce, militaries, medicine and life in general (Hymowitz, 2017):

> *Combined with faster processors, better sensors, and larger data sets, machine capabilities are growing exponentially. The data we now collect from everything from traffic sensors to Facebook visits to the Internet of things are essentially robot protein; more data, more powerful robots ... [implying] machines are becoming self-driving ...*

To be able to address disruption for coming and future systems, we want to see these questions addressed:

> *Will systems communicate with each other or be independent actors? Will humans always need to be in the loop? Will systems communicate only with human users, or also with robot and machine users? How will intelligent systems communicate?*

But there are few practical methods to address these questions for IoE systems. One proposal is to study IoT with agent-based models (ABMs). ABMs offer opportunities to pursue solutions to problems like IoT that happen to be too complex to solve by traditional methods, but ABMs are not yet ready (Houston et al., 2017):

> *Based on insights from this work, it is clear that this integration possesses great capacity for capturing the complexity of the modern world when compared to other forms of simulation and analysis. However, ... [ABMs are not yet capable of] answering practical business questions.*

When "things begin to think"

For the near future, we are becoming interested in what may happen when these "things begin to think". Foreseeing something like the arrival of the IoE, Gershenfeld (1999, p. 8, 10), the Director of MIT's Center for Bits and Atoms,[9] predicted that when a digital system

> *has an identity, knowing something about our environment, and being able to communicate ... components ... [must] work together ... so that the digital world merges with the physical world.*

Gershenfeld helps us to link this symposium with our past symposia in 2016 for using AI to reduce human errors[10] and another symposium in 2017 for using AI to determine "computational context", especially under uncertainty.[11] Gershenfeld leads us directly to intelligence.

Intelligence is a critical factor in overcoming barriers to direct maximum entropy production to solve difficult problems (Wissner-Gross & Freer, 2013; Martyushev, 2013). In battle systems, intelligence is necessary to

[3] WEC, 2015

[4] Kott et al., 2016

[5] Haghi et al, 2017

[6] Dibrov, 2017

[7] e.g., password authentication, back proofing, etc.

[8] To self-manage autonomy, human operators define the policies and rules to guide self-managed systems, identified by IBM as a self-* or self-star autonomous property (e.g., IBM, 2005).

[9] http://cba.mit.edu

[10] e.g., in 2016, AI and the mitigation of human error; see https://www.aaai.org/Symposia/Spring/sss16symposia.php#ss01

[11] e.g., our symposium on Computational context in 2017: https://aaai.org/Symposia/Spring/sss17symposia.php#ss03

complete missions by overcoming barriers, such as satisfying military "rules of engagement" (Mehta, 2017):

> *U.S. forces are no longer bound by requirements to be in contact with enemy forces in Afghanistan before opening fire, thanks to a change in rules of engagement orchestrated by Secretary of Defense Jim Mattis. Mattis ... told a pair of congressional hearings that the White House gave him a free hand to reconsider the rules of engagement and alter them to speed the battle against the Taliban if need be.*

But intelligence may also help to save humans lives. For example, a fighter plane can already take control and safe itself if its fighter pilot loses consciousness during a high-g maneuver.[12] We had proposed in 2016 that with existing technology, the passengers aboard Germanwings Flight 9525 might have been saved if the airliner had secured itself by isolating the copilot who purposively crashed his airliner to murder passengers and crew as he committed suicide (Lawless, 2016). Similarly, the Amtrak train that derailed in 2015 from the loss of awareness by its head engineer could have been spared the loss of life had the train slowed itself until it or its central authority had control of the train,[13] a remedy that might have prevented another train accident by an engineer in the State of Washington (Park & Yan, 2017):

> *Amtrak's president says the company is "profoundly sorry" after a train derailed this week in Washington state and hurtled off an overpass onto a freeway, killing three people. ... It's unclear why the train was traveling 80 mph in a 30-mph zone*

Gershenfeld's evolution may arrive when intelligent "things" and humans team together as part of a "collective intelligence" to solve problems and to save lives (Goldberg, 2017). A new theory on intelligence indicates that machine learning simulates compression and renormalization, both related to a lack of redundancy (Lawless, 2017). As reviewed by Wolchover (2017a),

> *Using [Shannon's] information theory ... Imagine X is a complex data set, like the pixels of a dog photo, and Y is a simpler variable represented by those data, like the word "dog." You can capture all the "relevant" information in X about Y by compressing X as much as you can without losing the ability to predict Y ... a deep-learning algorithm ... works ... as if by squeezing the information ... retaining only the features most relevant to general concepts ... like renormalization, a technique used in physics to zoom out on a physical system by coarse-graining over its details and calculating its overall state ...*

Relevant information is key when intelligent things replicate (Tegmark, 2017):

> *What's replicated isn't matter (made of atoms) but information (made of bits) specifying how the atoms are arranged. When a bacterium makes a copy of its DNA, no new atoms are created, but a new set of atoms are arranged in the same pattern as the original, thereby copying the information. ... [Similarly, human] synapses store all your knowledge and skills as roughly 100 terabytes' worth of information, while your DNA stores merely about a gigabyte, barely enough for a single movie download. ... even though the information in our human DNA hasn't evolved dramatically over the past 50 thousand years, the information collectively stored in our brains, books and computers has exploded.*

To better understand intelligence, England (2013) uses thermodynamic "fluctuation theorems" to quantify how humans select and shape certain physical processes happen than the reverse; e.g., from Wolchover (2017b),

> *by harvesting the maximum energy possible from the environment. Living creatures ... are superconsumers who burn through enormous amounts of chemical energy, degrading it and increasing the entropy of the universe ... groups of atoms that are driven by external energy sources ... tend to start tapping into those energy sources, aligning and rearranging so as to better absorb the energy and dissipate it as heat. ... Jeremy is showing ... that as long as you can harvest energy from your environment, order will spontaneously arise and self-tune*

Limitations

A possible limitation is that IoT "things" can be used to easily spy on users. James Clapper, the former US director of national intelligence, told the US Senate in public testimony (Thielman, 2016),

> *In the future, intelligence services might use the [IoT] for identification, surveillance, monitoring, location tracking, and targeting for recruitment, or to gain access to networks or user credentials,*

Privacy issues aside, the success of IoT depends on taming large quantities of data: 85% of all IoT devices are not yet connected and 4/5th of the data available is not yet structured for IoT; still, by 2020, machine data is expected to grow by 15 times; and stored data is expected to grow 50 times (Wind, 2015); e.g. (Fruehe, 2015),

[12] http://aviationweek.com/air-combat-safety/auto-gcas-saves-unconscious-f-16-pilot-declassified-usaf-footage

[13] https://www.nytimes.com/interactive/2016/05/17/us/amtrak-train-crash-derailment-philadelphia.html?_r=0

> *Companies today are grappling with the Internet of Things (IoT) ... encompassing devices, industrial equipment, sensors, and extended products. For some manufacturers everything they build could feed into IoT, from cars to buildings or even consumer products. ... Instead of focusing on the how of IoT, customers need to be focused on the what of IoT—namely the data. All of the strategy and shiny objects in the world won't help if the data isn't accurate, secure, and actionable. The data should always drive the strategy; the implementation tail should not be wagging the data dog.*

Yet, explaining the decisions made with machine intelligence is also a serious limitation; e.g., from Kuang (2017),

> *artificial intelligences often excel by developing whole new ways of seeing, or even thinking, that are inscrutable to us. It's a more profound version of what's often called the "black box" problem — the inability to discern exactly what machines are doing when they're teaching themselves novel skills — and it has become a central concern in artificial-intelligence research. ... [But] In 2018, the European Union will begin enforcing a law requiring that any decision made by a machine be readily explainable, on penalty of fines ...*

Further limiting machines are machine illusions; e.g., machine intelligence can be easily fooled (Somers (2017):

> *A deep neural net that recognizes images can be totally stymied when you change a single pixel, or add visual noise that's imperceptible to a human. Indeed, almost as often as we're finding new ways to apply deep learning, we're finding more of its limits. Self-driving cars can fail to navigate conditions they've never seen before. Machines have trouble parsing sentences that demand common-sense understanding ...*

More machine intelligence limitations were elaborated in an interview by Rodney Brooks, the famed roboticist at MIT (Miller, 2017; also, Garling, 2014) who stated that:

> *many of these detractors don't actually work in AI, and [he] suggested they don't understand just how difficult it is to solve each problem. "There are quite a few people out there who say that AI is an existential threat — Stephen Hawking, [Martin Rees], the Astronomer Royal of Great Britain ... they share a common thread in that they don't work in AI themselves ... For those of us who do work in AI, we understand how hard it is to get anything to actually work through [the] product level."*

Conclusion

For the Internet of Everything (IoE), in the future, we want to not only advance the present state of these "things" and to overcome its limitations, but also we want to manage how these "things" think so that the science of "collective intelligence" contributes to the welfare of society.

References

Chambers, J. (2014, 1/15), Are you ready for the Internet of everything? *World Economic Forum*, from https://www.weforum.org/agenda/2014/01/are-you-ready-for-the-internet-of-everything/

Cartwright, J.E., General (USMC, ret.), (2015, 11/10), "Leveraging the Internet of Things for a More Efficient and Effective Military", Cartwright is the Harold Brown Chair in Defense Policy Studies at the Center for Strategic and International Studies; he made his remarks at the Center for Strategic & International Studies (CSIS), from http://csis.org/event/leveraging-internet-things-more-efficient-and-effective-military

Dibrov, Y. (2017, December 18), he Internet of Things Is Going to Change Everything About Cybersecurity, Harvard Business Review, from https://hbr.org/2017/12/the-internet-of-things-is-going-to-change-everything-about-cybersecurity

England, J.L. (2013), Statistical physics of self-replication, J. Chem. Phys. 139, 121923 (2013); doi: 10.1063/1.4818538

Fruehe, J. (2015, 7/30), "The Internet Of Things Is About Data, Not Things", *Forbes*, from http://www.forbes.com/sites/moorinsights/2015/07/30/the-internet-of-things-is-about-data-not-things/#76d778cd74e4

Gasior, W. & Yang, L. (2013), Exploring covert channel in Android platform. 2012 International Conference on Cyber Security, pp. 173–177. DOI: 10.1109/CyberSecurity.2012.29

Garling, C. (2014, 12/24), "Smart" Software Can Be Tricked into Seeing What Isn't There. Humans and software see some images differently, pointing out shortcomings of recent breakthroughs in machine learning. MIT Technology Review, from http://www.evolvingai.org/files/MIT Tech Review Fooling paper.pdf

Gershenfeld, N. (1999), When things start to think. New York: Henry Holt & Co.

Goldberg, K. (2017, 6/11), "The Robot-Human Alliance. Call it Multiplicity: diverse groups of people and machines working together", Wall Street Journal, from https://www.wsj.com/articles/the-robot-human-alliance-1497213576

Haghi, M., Thurow, K., Habil, I. & Stoll, R (2017), Wearable Devices in Medical Internet of Things: Scientific Research and Commercially Available Devices, Healthcare Research Information, 23(1): 4–15, doi: 10.4258/hir.2017.23.1.4

Houston, C., Gooberman-Hill, S., Mathis, R., Kennedy, A., Li, Y. & Baiz, P. (2017), Case Study for the Return on Investment of Internet of Things Using Agent-Based Modelling and Data Science, Systems, 5(4): 1-46, doi:10.3390/systems5010004; from http://www.mdpi.com/2079-8954/5/1/4/pdf

Hymowitz, K.S. (2017, 7/18), "The Mother of All Disruption. Yes, robots are coming to the workplace, fast—and yes, America will change", City Journal, from https://www.city-journal.org/html/mother-all-disruptions-15251.html

IBM(2005), An architectural blueprint for autonomic computing, white paper, 3rd edition, from https://www-03.ibm.com/autonomic/pdfs/AC%20Blueprint%20White%20Paper%20V7.pdf

Kott, Alexander, Ananthram Swami, and Bruce J. West. "The Internet of Battle Things." *Computer* 49.12 (2016): 70-75.

Kuang, C. (2017, 11/21), "Can A.I. Be Taught to Explain Itself? As machine learning becomes more powerful, the field's researchers increasingly find themselves unable to account for what their algorithms know — or how they know it", New York Times, from https://www.nytimes.com/2017/11/21/magazine/can-ai-be-taught-to-explain-itself.html

Lawless, W.F. (2016), "Preventing (another) Lubitz: The thermodynamics of teams and emotion", in Harald Atmanspacher, Thomas Filk and Emmanuel Pothos (Eds.), Quantum Interactions. LNCS 9535, Springer International Switzerland, pp. 207-215.

Lawless, W.F. (2017), The physics of teams: Interdependence, measurable entropy and computational emotion, Frontiers physics. 5:30. doi: 10.3389/fphy.2017.00030

Marr, B. (2015, 10/27), "17 'Internet Of Things' Facts Everyone Should Read", *Forbes*, from http://www.forbes.com/sites/bernardmarr/2015/10/27/17-mind-blowing-internet-of-things-facts-everyone-should-read/#4893d3a01a7a

Martyushev, L.M. (2013), Entropy and entropy production: Old misconceptions and new breakthroughs, *Entropy*, 15: 1152-70.

Mehta, A. (2017, 10/3), "Mattis reveals new rules of engagement", Military Times, from https://www.militarytimes.com/flashpoints/2017/10/03/mattis-reveals-new-rules-of-engagement/

Miller, R. (2017, 7/25), "Artificial intelligence is not as smart as you (or Elon Musk) think", Tech Crunch, from https://techcrunch.com/2017/07/25/artificial-intelligence-is-not-as-smart-as-you-or-elon-musk-think/

Moskowitz, Ira S. (2017, 5/23), personal communication

Munro, K. (2017, 5/23), How to beat security threats to 'internet of things', from http://www.bbc.com/news/av/technology-39926126/how-to-beat-security-threats-to-internet-of-things

Park, M. & Yan, H. (2017, 12/21), "Amtrak train derailment leaves 'a thousand unanswered questions'", CNN, from http://www.cnn.com/2017/12/20/us/amtrak-derailment-washington/index.html

Ramachandran, S., Stevens, L. & Knutson, R. (2017, 7/6), "Amazon and Dish Network: A Match in the Making? A tie-up with the tech company could aid Dish Network's foray into wireless business", Wall Street Journal, from https://www.wsj.com/articles/amazon-dish-and-a-shared-vision-of-a-wireless-internet-of-things-1499333403

Somers, J. (2017, 9/29), "Is AI Riding a One-Trick Pony? Just about every AI advance you've heard of depends on a breakthrough that's three decades old. Keeping up the pace of progress will require confronting AI's serious limitations", MIT Technology Review, from https://www.technologyreview.com/s/608911/is-ai-riding-a-one-trick-pony/

Stankovic, J.A. (2014), Research Directions for the Internet of Things, IEEE Internet of Things Journal, 1(1): 3–9, DOI: 10.1109/JIOT.2014.2312291

Suri, N., Tortonesi, M., Michaelis, J., Budulas, P., Benincasa, G., Russell, S., ... & Winkler, R. (2016, May). Analyzing the applicability of internet of things to the battlefield environment. In Military Communications and Information Systems (ICMCIS), 2016 International Conference on (pp. 1-8). IEEE.

Tegmark, M. (2017, 8/29), "Will AI Enable the Third Stage of Life on Earth?. In an excerpt from his new book, an MIT physicist explores the next stage of human evolution", Scientific American, from https://blogs.scientificamerican.com/observations/will-ai-enable-the-third-stage-of-life-on-earth/

Thielman, S. (2016, 2/10), "The internet of things: how your TV, car and toys could spy on you. As our homes get 'smart', the US intelligence chief has said the data involved could be used for surveillance. Here's how that could affect us all", *The Guardian*, from http://www.theguardian.com/world/2016/feb/10/internet-of-things-surveillance-smart-tv-cars-toys

WEC (2015, January), Industrial Internet of Things: Unleashing the Potential of Connected Products and Services, World Economic Forum (In collaboration with Accenture), from http://www3.weforum.org/docs/WEFUSA_IndustrialInternet_Report2015.pdf

Wind (2015), "The internet of things for defense", Wind, an INTEL Company White Paper, from http://www.windriver.com/whitepapers/iot-for-defense/wind-river_%20IoT-in-Defense_white-paper.pdf

Wissner-Gross, A. D., and C. E. Freer (2013), Causal Entropic Forces, Physical Review Letters: 110(168702): 1-5.

Wolchover, N. (2017a, 9/21), "New Theory Cracks Open the Black Box of Deep Learning. A new idea called the "information bottleneck" is helping to explain the puzzling success of today's artificial-intelligence algorithms — and might also explain how human brains learn", Quanta Magazine, from https://www.quantamagazine.org/new-theory-cracks-open-the-black-box-of-deep-learning-20170921/

Wolchover, N. (2017b, 7/26), "First Support for a Physics Theory of Life. Take chemistry, add energy, get life. The first tests of Jeremy England's provocative origin-of-life hypothesis are in, and they appear to show how order can arise from nothing", Quanta Magazine, from https://www.quantamagazine.org/first-support-for-a-physics-theory-of-life-20170726/

Active Inference in Multi-Agent Systems: Context-Driven Collaboration and Decentralized Purpose-Driven Team Adaptation

Georgiy Levchuk, Krishna Pattipati, Daniel Serfaty, Adam Fouse, Robert McCormack

Aptima Inc., 12 Gill St., Suite 1400, Woburn, MA 01801, USA
georgiy@aptima.com

Abstract

Internet of things (IoT), from heart monitoring implants to home heating control systems, are becoming an integral part of our daily lives. We expect these technologies to become smarter, able to autonomously reason, act, and communicate with other entities in the environment and act to achieve shared goals. To realize the full potential of these systems, we need to understand the mechanisms that allow multiple agents to effectively operate in changing and uncertain environments. This paper presents a framework that postulates that optimal multi-agent systems achieve adaptive behaviors by *minimizing the team's free energy*, where energy minimization process consists of incremental *perception* (inference) and *control* (action) phases. We discuss instantiation of this mechanism for a problem of joint distributed decision making, provide the concomitant abstractions and computational mechanisms, and present experimental evidence that energy-based agent teams significantly outperform utility-based teams. We discuss different adaptation mechanisms and scales, explain agent interdependencies produced by energy-based modeling, and look at the role of *learning* in the adaptation process. We hypothesize that to efficiently operate in uncertain and changing environments, IoT devices must not only maintain enough intelligence to perceive and act locally, but also possess team-level adaptation primitives. We posit that such primitives must embody energy-minimizing mechanisms but can be locally defined without the need for agents to possess global team-level objectives or constraints.

Introduction

Autonomous intelligent systems are no longer a fancy of science fiction writers, but quickly becoming part of our everyday life. These devices, from heart monitoring implants to home heating control systems, have been designed to make our lives easier. Many commercial technology companies are racing against each other to bring new devices to the market, making them "smarter" every day. While most of the currently deployed internet of things (IoT) systems perform simple tasks, such as providing environment monitoring and human-guided control for smart homes, hospitals, or assembly plants, it is not difficult to envision a near future in which the intelligence and authority of these devices expand beyond their current applications.

Most previous research in the area of IoT intelligence focused on hardware-software interoperability and human-machine interfaces (Al-Fuqaha et al., 2015), standards and architectures for contextual reasoning (Perera et al., 2015), and operational challenges for a single device or networks of homogeneous IoT components (Whitmore, Agarwal, and Da Xu, 2015). As smart devices interact with each other, the human users, and the data, the implications to this internet of everything (IoE) (Evans, 2012) must be thoroughly studied. This would allow the development of models to extract the highest potential from multiple autonomous and heterogeneous intelligent systems, including human-machine teaming identified as the defense technology of the future (Pellerin, 2015).

In this paper, we address two fundamental issues in IoE. First, we describe a **general framework of adaptive multi-agent behavior** based on *minimizing the team's free energy*. This framework explains how multiple intelligent agents can produce team-optimal context-aware behaviors by performing collaborative perception and control. Second, we present a mechanism for IoE agents to **instantiate adaptive behaviors** by *intelligently sampling their environment* and *changing their organizational structure*. This structure specifies roles and relations that encode and constrain the agents' responsibilities and interactions, and can be modified in a collaborative and distributed manner by the agents themselves without a central authority. Energy-based team adaptation formally connects the concepts of local and global perceptions, decisions, and knowledge, while local-global synchronization can be achieved using peer-to-peer communications. We show how this model provides a mathematically principled mapping between the adaptation, decentralized purpose-driven behaviors, and perception in multi-agent systems, and prescribes foundational functional requirements for developing IoE networks that can efficiently operate in complex, dynamic, and uncertain environments of the future.

Energy-based Adaptive Agent Behaviors

Free Energy Principle

Recently, a theory has been proposed that suggests that agents, e.g., biological systems such as a cell or a brain, adapt to their environments by reducing the information-theoretic quantity known as "variational free energy" (Friston, 2010). This can also be reinterpreted to mean that agents reduce their free energy by adapting to their environment (Friston, Thornton, and Clark, 2012). This theory, called "free energy principle", brings information-theoretic, Bayesian, neuroscientific, and machine learning approaches into a single framework, by formalizing that agents can reduce the free energy in three ways: (i) by changing sensory input (*control*); (ii) by changing predictions of the sensor inputs (*perception*); and (iii) by changing the model of the agent such as its form and structure (*learning*).

Variational free energy is defined as a function of sensory outcomes and a probability density over their (hidden) causes. This function is an upper bound on *surprise*, a negative log of the model evidence representing the difference between agent's predictions about its sensory inputs, and the observations it actually encounters. Since the long-term average of surprise is entropy, an agent acting to minimize free energy will implicitly place an upper bound on the entropy of the outcomes – or sensory states – it samples. Consequently, free energy principle provides a mathematical foundation to explain how agents maintain their order by restricting themselves to a limited number of states. This gives a formal mechanism to design decentralized purpose-driven behaviors, where multiple agents can operate autonomously and resist disorder without control by any external agent.

Adaptive Behavior and Context

Free energy generalizes to the case of learning and cognition, prescribing that acquisition of any form of knowledge can be viewed as an attempt to reduce surprise. Moreover, a fundamental property of this formulation, as can be seen below from its mathematical derivations, is that both free energy and the surprise it bounds are highly contextual. First, surprise is a function of sensations and the agent predicting them, and thus exists only in relation to model-based expectations. Surprise-minimizing agents are attempting to adapt themselves to the context gathered from their observations. Second, free energy principle suggests that agents harvest sensory signals they can predict, and keep to consistent subspaces of all physical and physiological variables that define their existence (Friston, Thornton, and Clark, 2012). Under constant perception about the world, the energy-minimizing agent would change its actions to maximize the entropy of the sensations (and, accordingly, self-information) it receives. In other words, the adaptive behaviors prescribed by free energy principle tightly couple the environment and the agents that populate it and conform to those behaviors.

In addition, the minimization of surprise suggests that the adaptive action selection cannot be deterministic. This provides a key differentiation between the behaviors based on free energy principle, and the classical control formulations in which utility or some cost functions are optimized. Essentially, the contextual information encoded by the free energy produces stochastic actions to achieve a boundedly rational behavior.

It should be noted that similar ideas have been pursued in manual control and normative-descriptive models of human decision making. For example, in manual control, it is hypothesized that a well-trained and well-motivated human acts optimally to control a system, subject to his/her perceptual and neuro-motor limitations and the perceived task objectives (e.g., Kleinman, Baron, and Levison, 1971). Indeed, the assumption that every good regulator/controller of a system must be a model of that system is inherent to the Internal Model control (IMC) principles of control theory (Smith, 1959; Conant, and Ashby, 1970). The basic assumption underlying human decision making in dynamic contexts is that a well-trained human behaves in a normative, rational manner subject to his inherent biases and limitations (e.g., Pattipati, Kleinman, and Ephrath, 1983). The concept of surprise (innovation, residuals) is the basis for modern estimation theories, system identification, anomaly detection and adaptive control (e.g., Bar-Shalom, Li and Kirubarajan, 2001; Ljung, and Glad, 1994).

Formal Definitions

Given an agent and its generative model of environment m, we can formally model purpose-driven adaptive systems as "behaving rationally" by maximizing model evidence, a probability distribution $p(o|m)$ over observations o conditioned on model of the environment m, or minimizing the measure of surprise:

$$surprise(o,m) = -\ln p(o|m).$$

However, direct optimization of model evidence or surprise is intractable, because it requires marginalization over all possible hidden states of the world (Friston, 2012). Recently, researchers conjectured that the only tractable way to optimize surprise is to minimize the variational free energy $F(o,b)$, an information-theoretic function of outcomes o and an internal state of the agent defined as a probability density b over (hidden) causes of these outcomes (Friston, Thornton, and Clark, 2012):

$$F(o,b) = \underbrace{E_q[-\ln p(s,o|m)]}_{average\ energy} - \underbrace{H[q(s|b)]}_{entropy},$$

where:

- $p(s,o|m)$ is a *generative density* representing joint probability of world states s and observations o based on an agent model m;
- $q(s|b)$ is a *recognition density* that defines agent's beliefs about hidden states s given internal state of agent b;
- $E_q[\cdot]$ is the expected value over recognition density, i.e. $E_q[-\ln p(s,o|m)] = -\sum_s q(s|b) \ln p(s,o|m)$; and
- $H[\cdot]$ is the entropy of the recognition density, i.e. $H[q(s|b)] = -\sum_s q(s|b) \ln q(s|b)$.

By rewriting free energy function, we can obtain several interpretations of how adaptive agents "behave". First, free energy is equal to the sum of surprise and divergence, obtaining that free energy is an **upper bound on surprise**:

$$F(o,b) = \underbrace{-\ln p(o|m)}_{surprise} + \underbrace{D_{KL}[q(s|b) || \, p(s|o,m)]}_{divergence}$$
$$\geq -\ln p(o|m)$$

In the above, $D_{KL}[q(s|b) || \, p(s|o,m)]$ is the Kullback-Leibler divergence between the recognition density $q(s|b)$ and the true *posterior* of the world states $p(s|o,m) = p(s|o)$. Consequently, minimization of free energy achieves approximate minimization of surprise, and is achieved when the perceptions $q(s|b)$ are equal to the posterior density $p(s|o,m)$.

Second, we can rewrite the free energy as the difference between complexity and accuracy:

$$F(o,b) = \underbrace{D_{KL}[q(s|b) || \, p(s|m)]}_{complexity} - \underbrace{E_q[-\ln p(o(a)|s,m)]}_{accuracy}$$

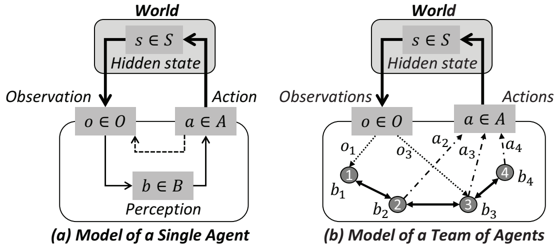
Here, $D_{KL}[q(s|b) || \, p(s|m)]$ is a measure of divergence between the recognition density $q(s|b)$ and *prior* beliefs about the world $p(s|m)$, and thus can be interpreted as a measure of complexity, while the second component is the expectation about the observations o to be received after performing an action a, which represents accuracy. This means that the agent modifies the sensory outputs $o = o(a)$ through action a to achieve the most accurate explanation of data under fixed complexity costs. Accordingly, we can now define the free energy minimization using two sequential phases that separate estimation and control:

- *Perception* phase finds beliefs $b^* = \arg\min_b F(o,b)$; and
- *Control* phase finds actions $a^* = arg\,\min_a F(o(a), b^*)$.

The control phase produces a policy for the agent that will generate observations that entail, on average, the smallest free energy. This ensures that the individual actions produced over time are not deterministic, and the control phase can be converted into a sampling process $a^* \sim Q(a,b,o)$ using a function of exploration bonus plus expected utility (Friston et al., 2013) or average free energy (Friston, Samothrakis, and Montague, 2012). Further, the free energy is dependent on the agent's model m, which can be adapted to minimize free energy via evolutionary or neuro-developmental optimization. This process is distinct from perception, and entails changing the form and architecture of the agent (Friston, Thornton, and Clark, 2012). This means that free energy function can be used to compare two or more agents (models) to each other (a better agent is the one that has the smaller free energy), and thus is an ultimate measure of *fitness*, or *congruence* or *match* between the agent and its environment.

Behavior Workflow and Computational Considerations

Fig. 1 depicts a simplified schematic of the resulting cycle of sensing, control and perception in adaptive agents, where posterior expectations (about the hidden causes of observation inputs) minimize free energy and prescribe actions. The team of agent differs from a single agent model by requiring the observations, perceptions and actions be distributed among multiple agents, while allowing the agents to communicate to achieve team-level goals.



(a) Model of a Single Agent

(b) Model of a Team of Agents

Figure 1: Schematic of the interdependencies among variables of adaptive agent & team models.

The main benefit of using information-theoretic free energy principle for modeling dynamical systems is that the function $-\ln p(s,o|m)$ can have a simple mathematical structure when the generative density $p(s,o|m)$ factors out:

$$p(s,o|m) = \frac{1}{Z} \prod_i \varphi_i(\boldsymbol{s}_i, \boldsymbol{o}_i),$$

where $\{\boldsymbol{s}_i, \boldsymbol{o}_i\}$ represent the subsets of state and observation variables, φ_i are factor functions encoding dependency relations among the corresponding variables, and Z is the normalization constant. Usually, functions $\varphi_i(\cdot)$ are simple, typically describing the relations among one to four variables at a time. In this situation, the energy minimization with respect to internal state, i.e. recognition density $q(s|b)$, can be performed using generalized belief propagation (BP) algorithm (Friston et al., 2013), an iterative procedure based on message passing. Moreover, using a standard BP algorithm, which is derived from a Bethe approximation to the variational free energy (Yedidia, Freeman, and Weiss, 2005), we can obtain the approximating density in a peer-to-peer manner and with low computational complexity. While standard BP does not guarantee convergence, it performs quite well in practice.

The free energy principle does not dictate the specifics of a generative process, i.e. the structure/components of the generative density $p(s,o|m)$ required to define the free energy function. Nor does it prescribe the algorithms that need to be employed to minimize free energy. However, it provides unifying framework and can be tailored to specific environments and systems. Here, we proceed to apply the free energy formalisms to the design of adaptive multi-agent teams, defining appropriate abstractions, and discussing their implications for the IoE functional requirements.

Application of Energy Formalism to Multi-Agent Teams

Motivation

A team consisting of human and machine agents is a decentralized purpose-driven system. One of the main challenges in defining adaptive team behavior is one of realizing global team-level perception and control, and corresponding optimization processes, into local inferences and decisions produced by individual agents without external control.

In our previous work, we showed how free energy minimization can be applied to define adaptive behavior in teams that execute given multi-task missions (Levchuk et al., 2017). Examples of such teams include military organizations, manufacturing teams, and many other project-based organizations. These teams interact with their environment by jointly assigning and executing tasks, and teams with the highest task execution accuracy and/or the fastest execution time are considered more efficient. In the domain of project-based teams, we considered team members (agents) to possess high levels of intelligence, and thus assumed that the agents' task assignment processes cannot be directly controlled. We thus defined observations as the outcomes of task execution, and treated task assignments as hidden world state variables. Consequently, the perception phase estimated the probability of agent-to-task assignment, while the control phase defined the team's organization structure, which included the roles and relations that constrained what tasks the agents could execute. The team structure restricted the assignments the agents could generate, thus formally providing a process for the team to resist the disorder.

In this paper, we discuss an instantiation of the adaptive behavior in a distributed decision-making setting. This is a more general setup than project-based teams, and was motivated by the following. First, we wanted to understand how the intelligent adaptive behaviors are related to the formal dynamics and structures among agents. This we set out to do by studying how organizational structure impacts the process of searching for good sets of decisions and the process stabilizes around good decisions once they are discovered (Rivkin and Siggelkow, 2003). The search and stability issues are conceptually identical to the exploration-exploitation tradeoffs afforded by free energy minimization, and we wanted to examine the alignment between energy-based computational mechanisms and discrete human decision-making processes analyzed in (Rivkin and Siggelkow, 2003).

Second, we posit that free energy could explain the empirically observed behaviors of business organizations. However, unlike network-based theory of cities (Schläpfer et al., 2014), where extensive quantitative data about social interactions is available, data on operations of business enterprises (e.g., all communication channels, personnel assignments, and task outcomes) is not available, and this so far has prevented the development of a mechanistic framework for organizations. It has been empirically observed, however, that as the companies mature and grow, they attempt to maximize profits (utility) at the expense of innovation (entropy or disruption or disorder), placing increasing emphasis on rules, regulations, and other forms of bureaucratic control over its members, leading to their eventual demise (West, 2017).

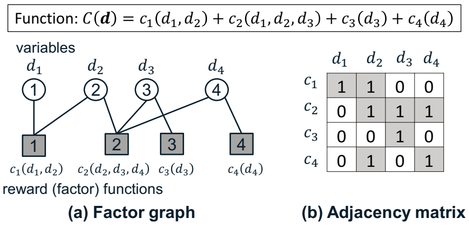
Finally, we wanted to identify what implications the free energy minimization principle has on designing agents that constitute effective members of a high-performance team.

Problem Definition

The formal definition of a distributed decision making problem given in (Rivkin and Siggelkow, 2003) is as follows. Assume that a team of M agents needs to find an N-dimensional binary decision vector $\boldsymbol{d} = [d_1, \dots, d_N]$, where $d_i \in \{0,1\}$, to maximize the additive objective function, i.e.:

$$\boldsymbol{d}^* = \underset{\boldsymbol{d}\in\{0,1\}^N}{\arg\max}\, C(\boldsymbol{d}) = \sum_{j=1}^{P} c_j(\boldsymbol{d}_j),$$

where $\boldsymbol{d}_j$ is a subset of decision variables, and P is the number of these subsets. We assume that the component reward functions $c_j(\boldsymbol{d}_j)$ encode dependencies between the decisions in the subset $\boldsymbol{d}_j$ (such as local and team-level rewards), and are randomly chosen. The general additive form of the objective function $C(\boldsymbol{d})$ can represent many real-world search and inference problems. Relationships among the reward functions and decision variables can be expressed using the factor graph or function-to-decision adjacency matrix (Fig. 2).


Function: $C(\boldsymbol{d}) = c_1(d_1, d_2) + c_2(d_1, d_2, d_3) + c_3(d_3) + c_4(d_4)$
variables
d_1 d_2 d_3 d_4
1 2 3 4
1 2 3 4
$c_1(d_1, d_2)$ $c_2(d_2, d_3, d_4)$ $c_3(d_3)$ $c_4(d_4)$
reward (factor) functions

	d_1	d_2	d_3	d_4
c_1	1	1	0	0
c_2	0	1	1	1
c_3	0	0	1	0
c_4	0	1	0	1

(a) Factor graph **(b) Adjacency matrix**

Figure 2: Example of defining distributed decision-making problem using a factor graph and adjacency matrix

As the number N of decision variables increases, the space of possible points grows exponentially[1]. As a result, the optimal solution to the above maximization problem cannot be achieved by exhaustively looking up the values of joint reward function $C(\boldsymbol{d})$. Instead, an intelligent search in the space of all decisions needs to be conducted by the agents. To constrain what the agents can do and enable their collaboration, the organizational structure m among the agents is defined using two variables (Fig. 3):

- *decision decomposition*, prescribing subsets of decisions and cost functions assigned to each agent; and
- *agent network*, prescribing superior-subordinate relations among the agents.

The agents are locally aware of and control only the subsets of decisions and reward functions assigned to them, giving rise to local-global decision inconsistency. Accordingly, to produce team-optimal decisions (those that maximize the team objective function $C(\boldsymbol{d})$), agents need to collaborate.

In (Rivkin and Siggelkow, 2003), a version of decision-making and collaboration process was defined to mimic human decision-making in business organizations. The subordinate agents generate a set of discrete decision vectors, rank these vectors using local payoffs, and communicate the ranked vectors to a superior agent (indicated as *CEO* in Fig. 3b). The superior agent combines the subordinates' vectors into a set of candidate team-level decision vectors, evaluates them against the team's objective function, and communicates the best vector back to the subordinates for implementation. The number of decision vectors considered by the agents is constrained by their internal capacity, specified by a bound on the number of alternatives the agent can evaluate in a limited amount of time. Lateral relationships can also be defined among the agents (Siggelkow and Rivkin, 2005), allowing one agent to inform another about their local decisions on their own interdependent variables.

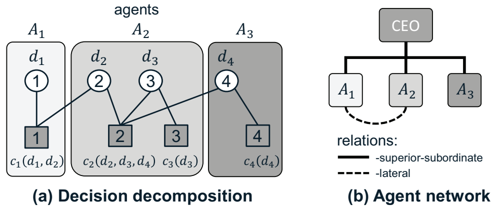


Figure 3: Variables of team structure

The decision-making process described in (Rivkin and Siggelkow, 2003) presents a heuristic solution to a distributed optimization problem, with no guarantees of convergence or efficacy even with the introduction of global incentives for the agents. We can assess the "behavior" of a multi-agent team using these heuristics in terms of the time it takes to produce a solution, its proximity to the true maximum, or the amount of exploration vs exploitation in the state space of all decision vectors the agents jointly generate. However, this heuristic does not explain the causes of the underlying behavior, nor prescribe the adaptive behaviors for the team or its members. In the following, we provide a formal optimization process to solve this problem in a distributed way using the free energy minimization principles, and describe the concomitant adaptive team behavior exhibited by the team members during the search process. We show that energy-based search significantly outperforms the discrete optimization heuristics in (Rivkin and Siggelkow, 2003), and provides a mechanism to describe adaptive collaborative behaviors.

Distributed Collaborative Search via Free Energy Minimization

The problem described above can be recast as a joint inference over a factorized objective function in Section 2.4. Since the environment is fully observable (i.e., the observation o is equal to the decision vector $\boldsymbol{d}$) in the distributed search problem of Section 3.2, we omit the observation notation from the rest of the exposition. We postulate that the adaptive behavior based on free energy principle for a team of agents can be described using three processes:

- *Perception* will find the beliefs b representing the probability distribution $q(\boldsymbol{d}|m)$;
- *Control* will produce the next decision $\boldsymbol{d}$ by sampling the state space to minimize surprise; and
- *Reorganization* will adapt the structure m among agents in terms of decision decomposition and agent network.

Formally, we define a generative probability distribution for a decision variable $\boldsymbol{d}$ as:

$$p(\boldsymbol{d}|m) \cong \frac{1}{Z} e^{C(\boldsymbol{d})} = \frac{1}{Z}\prod_{j=1}^{P} \varphi_j(\boldsymbol{d}_j),$$

[1] This is a so-called NP-hard problem, meaning that there are no known polynomial-time algorithms for this problem.

where $\varphi_j(\boldsymbol{d}_j) = e^{c_j(\boldsymbol{d}_j)}$. We can then write the variational free energy as a function of beliefs b and of the team structure m:

$$F(b, m) = E_q[-\ln p(\boldsymbol{d}|m)] - H[q(\boldsymbol{d}|b)].$$

Minimizing the variational free energy $F(b, m)$ with respect to probability functions $q(\boldsymbol{d}|b)$ is therefore an exact procedure for bounding surprise and recovering $p(\boldsymbol{d}|m)$. However, exact minimization is intractable for general forms of $p(\boldsymbol{d}|m)$ due to curse of dimensionality.

When the generative probability is factorizable, as described above, generalized belief propagation can be used to find the marginal probability distributions (Yedidia, Freeman, and Weiss, 2005), which form the basis for generating the next decision points stochastically. However, this method incurs high computational cost involving iterative update equations, and is difficult to design with respect to appropriate clusters of decisions and factor functions, defining agents; this also affects the speed of convergence.

Two features help us tackle these challenges. First, we note that maximizing the team cost function is equivalent to maximizing the joint decision probability function. This process, also known as maximum a-posteriori (MAP) estimation, requires obtaining max-marginal probability values, rather than marginal probabilities. Second, instead of exact computation of belief distributions, we use an approximate solution produced by the standard belief propagation algorithm (Yedidia, Freeman, and Weiss, 2005), which is based on Bethe approximation of the free energy function. Hence, we use the max-product algorithm, reducing the space of distributions to analyze, thereby lowering the computational complexity. Max-product belief propagation algorithm computes max-marginal distributions by iteratively passing belief messages between variable and factor nodes in the factor graph (Fig. 4a) with the following update equations:

- *Variable-to-factor* message updates involve multiplication of all except one of the incoming beliefs:

$$m_{i\to j}(d_i) = \prod\nolimits_{c \in N(i)\backslash j} m_{c\to i}(d_i)$$

- *Factor-to-variable* message updates are conducted by maximizing the component functions*:*

$$m_{j\to i}(d_i) = \max_{\boldsymbol{d}_j \backslash d_i} \varphi_j(\boldsymbol{d}_j) \prod\nolimits_{v \in N(j)\backslash i} m_{v\to j}(d_v)$$

In the above, $N(i)$ denotes the neighbor nodes of node i in factor graph. For a team of agents, some of the computations above are conducted locally by a single agent, while the message passing across the links between variables and factors that are assigned to different agents need to be physically passed among the agents (Fig. 4b). These messages thus define a formal local-global decision making, while the collaborative process is formally specified as the belief messages sent between connected agents.

Variable-to-factor messages contain the beliefs about the variable, and thus can be conceptualized as **experience messages**. Factor-to-variable messages attempt to change the probability distribution at the variable node, and thus can be termed **influence messages**. Thus, the free energy principle and approximate inference using max-product algorithm only require the agents to understand the dependencies of their local decisions on the decisions of other agents, and to be capable of sharing and interpreting the experience and influence messages.

After max-product algorithm converges, the agents compute the max-marginal probability distributions, and use them to sample the space of the decision vectors locally (i.e., each agent samples its own subset of decision vector variables). Max-marginal estimates include variable marginals, which agents use to sample the decision space:

$$b_i(d_i) = \max_{\boldsymbol{d}\backslash\{d_i\}} p(\boldsymbol{d}|b) \propto \prod\nolimits_{j \in N(i)} m_{j\to i}(d_i),$$

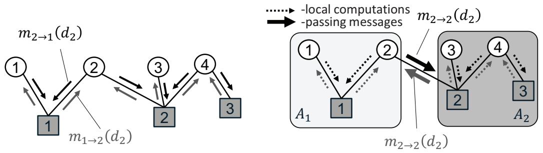
as well as factor marginals, which are used to adapt the team structure:

$$b_j(\boldsymbol{d}_j) = \max_{\boldsymbol{d}\backslash\{\boldsymbol{d}_j\}} p(\boldsymbol{d}|b) \propto \varphi_j(\boldsymbol{d}_j) \prod\nolimits_{i \in N(j)} m_{i\to j}(d_i)$$

Using above quantities, we can compute so called *Bethe approximation* to the free energy function (Yedidia, Freeman, and Weiss, 2005):

$$F_{Bethe}(b, m) = E_{Bethe}(b, m) - H_{Bethe}(b, m),$$

in which the first component is negative expected utility computed as $E_{Bethe}(b, m) = -\sum_{\boldsymbol{d}_j} b_j(\boldsymbol{d}_j) c_j(\boldsymbol{d}_j)$, and the second component is the entropy $H_{Bethe}(b, m) = (n_i - 1)\sum_i \sum_{d_i} b_i(d_i) \ln b_i(d_i) - \sum_j \sum_{\boldsymbol{d}_j} b_j(\boldsymbol{d}_j) \ln b_j(\boldsymbol{d}_j)$ (where $n_i = |N(i)|$ is the number of factors $\boldsymbol{d}_j$ the variable i is involved in). This means that minimizing free energy is achieved when the team finds all possible (maximally varying) marginals with highest utility.



(a) Passing beliefs in factor graph (b) Collaboration by message passing

Figure 4: Message passing in belief propagation

Adapting Team Structure

Team structure (model) m affects the decisions the team jointly produces. Indeed, the structure impacts how the information flows in the organization (i.e., the message passing processes, decision sampling in our formulation) and the concomitant workload incurred by team members (agents), message delays and message transmission errors. This problem of team structuring can be formulated (and solved) as a network design problem (Feremans, Labbé, and Laporte, 2003).

The decision decomposition and the corresponding BP message calculations represent the *internal computational workload* incurred by the agents, and represent the collaboration process required to solve the decision problem. Specification of messages to be passed between decision and function nodes in a factor graph define the *external communication workload* of the agents. Thus, the problem of structuring a team can be formulated as the alignment of decision decomposition ("the task network") and the agent network ("team structure") by properly balancing the internal and external workloads. We model the impact of internal and external workloads on the team's perception and action processes using delay models and/or message transmission errors (the classic "delay-accuracy" tradeoff in team performance). Schematically, the team structuring problem can be decomposed in two steps (Fig. 5). First, we aggregate the decision and factor nodes to balance the internal and external workloads via cluster analysis by minimizing the inter-cluster links, subject to a constraint on the workload capacities of agents. Second, we use this network as a set of requirements, and map it to the current agent network, re-aligning agent network parameters (such as capacities) as the factor graph evolves (e.g., changes in decision reward parameters).

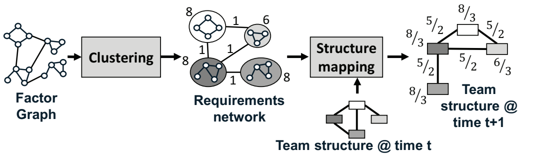


Figure 5: Team structure adaptation workflow

Validation Experiments

Experiment Setup

To validate our proposed adaptive team behavior model, we conducted several studies. First, we generated a collection of synthetic distributed decision-making problems by randomly generating the values of joint reward function. We manipulated the density of the dependencies between variables using a parameter *K*, defined as the average number of variables influencing each factor (reward) node. We computed several assessment metrics, including percent of trials where the team converged to the optimal solution (determined as producing a vector with the reward value within a small threshold of the optimal value), and normalized payoff (computed as a fraction of the obtained solution to the optimal one). We evaluated team performance over time, as well as on time averaged payoff. Second, we introduced random periodic variations of the parameters of the joint reward functions and analyzed how quickly the teams can recover from these changes. The latter analysis enables us to quantify the attributes of resilience in a team: (i) capacity to absorb changes, (ii) recoverability, and (iii) adaptive capacity (Francis and Bakera, 2014).

Using the generated datasets, we conducted several evaluations, comparing discrete decision-making model of (Rivkin and Siggelkow, 2003) with two alternative decision processes: perception-maximizing decisions (selecting decisions as maximum of max-marginal probabilities), and energy minimization (selecting decision by sampling using max-marginals, which minimizes surprise). We also analyzed the impact the specific team structures (models) had on the ability of the team to find correct solutions.

Discrete Decision-Making vs Free Energy

In our first evaluation, we compared the performance of distributed discrete decision making (D3M) heuristics with our model, where the decision vector is selected by minimizing free energy. Fig. 6 shows the normalized payoff achieved at 100th iteration by the best of D3M policies described in (Rivkin and Siggelkow, 2003), and one of the teams defined by our method. While free energy solution provides only marginal improvement at this point of the simulation, our model achieves convergence much faster than D3M heuristics (usually at 15 iterations vs 50-80 iterations for D3M), and maintains high convergence for increasing objective function complexity (parameter *K*), while the performance of D3M consistently decreases with *K*. Therefore, we continued to compare only energy-based adaptation processes.

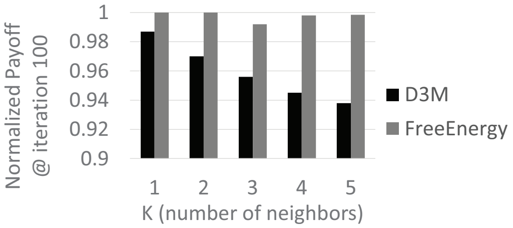


Figure 6: Comparing performance of D3M heuristic vs free energy minimization.

In the next set of experiments, we analyzed the ability of the team to adapt to changes in the environment, which we defined as random regeneration of objective function's parameters (without changing the topology of the factor graph) introduced at every 20 decision iterations.

Impact of Agent Network Structure

First, we compared the effects of three different agent network topologies (Fig. 7) on the quality of the search process and the ability of the team to adapt. Here, we also analyzed different behaviors of the root node (CEO) in the hierarchy: "active" refers to the agent that passes indirect messages among the subordinate agents, and "passive" accounts for ignoring those messages completely.

From the average payoff values in Fig. 8, we conclude that organizations with stronger subordinates ("Lateral" and "Fully Connected") perform better, while the relative benefit of such teams is highest for medium dependencies between decisions (*K*=2-3). In these situations, the benefit of lateral coordination appears to outweigh the cost of managing multiple communications. The relative benefit of fully connected networks reduces as the dependencies become more complex (*K*>3), mostly due to suboptimality of the distributed solution when there are many dependencies (i.e., large *K*) among the decision variables.

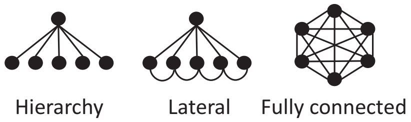


Figure 7: Considered team structures

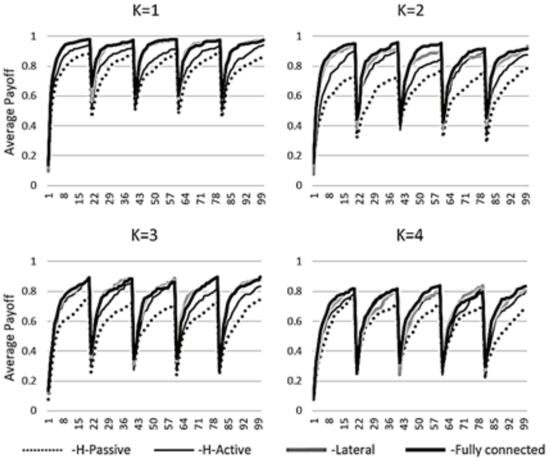


Figure 8: Comparison of effect of team structure on average payoff (H-Active/Passive = hierarchy with active or passive CEO).

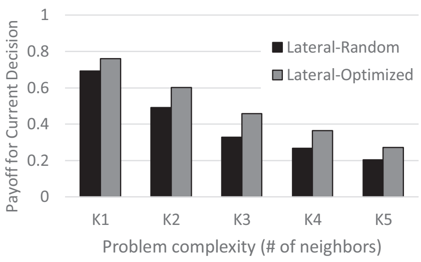


Figure 9: Example of effect of decision decomposition on the quality of current solution for a lateral team structure.

Impact of Decision Decomposition

Finally, we studied the effect of decision decomposition (assignment of decision and factor nodes to agents) on team performance. We computed a score on the quality of current decision as the value of objective function at the max-marginal vector $\widehat{\boldsymbol{d}} = \{\hat{d}_i\}$, where

$$\hat{d}_i = \arg\max b_i(d_i).$$

We found that using optimized vs random decomposition improved the solution achieved by the team, with the larger effect for lateral structure (Fig. 9).

Due to space limitations, we omitted analysis of (a) how team structures affect performance; (b) correlation between free energy and the reward function improvement; (c) internal/external workload metrics and how they impact the decision quality, (d) measures of resilience. These will be included in a future publication.

Conclusions

In this paper, we studied the problem of generating adaptive behaviors for cooperating agents. We presented an application of free energy minimization principle to generate decentralized purpose-driven teams of agents. Experiments with synthetic data prove that energy-based behavior results in higher performance on a distributed search task compared to discrete decision making heuristics. Minimum free energy formalism provides a mathematically sound mechanism for coupling perception and action selection processes. Finally, the decisions to affect the environment though action, adapt by modifying perception, and adjust the architecture of a team in terms of the organizational structure among the agents can all be executed in a distributed collaborative manner, without the need for an external controlling agent.

One of the key innovations of our work is that it prescribes two general interfaces the intelligent adaptive agents must possess: generating, communicating, and incorporating the experience messages and the influence messages. Our current work is focused on defining precise free energy functional that encodes effects of team structure of decisions and communications, studying the convergence properties of distributed perception and control processes, obtaining the collaborative adaptation mechanisms for project-based teams, and deriving high-level corollaries and general trends from lower-level free energy minimizing processes.

References

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. and Ayyash, M., 2015. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4), pp.2347-2376.

Bar-Shalom, Y., Li, X. R., & Kirubarajan, T. (2004). *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons.

Conant, R. C., & Ross Ashby, W. (1970). Every good regulator of a system must be a model of that system. *International journal of systems science*, 1(2), 89-97.

Evans, D., 2012. The internet of everything: How more relevant and valuable connections will change the world. *Cisco IBSG*, pp.1-9.

Feremans, C., Labbé, M. and Laporte, G., 2003. Generalized network design problems. *European Journal of Operational Research*, 148(1), pp.1-13.

Francis, R., & Bekera, B. (2014). A metric and frameworks for resilience analysis of engineered and infrastructure systems. *Reliability Engineering & System Safety*, 121, 90-103.

Friston, K., 2010. The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2), pp.127-138.

Friston, K., Thornton, C. and Clark, A., 2012. Free-energy minimization and the dark-room problem. *Frontiers in psychology*, 3.

Friston, K., 2012. A free energy principle for biological systems. *Entropy*, 14(11), pp.2100-2121.

Friston, K., Samothrakis, S. and Montague, R., 2012. Active inference and agency: optimal control without cost functions. *Biological cybernetics*, pp.1-19.

Friston, K., Schwartenbeck, P., FitzGerald, T., Moutoussis, M., Behrens, T. and Dolan, R.J., 2013. The anatomy of choice: active inference and agency. *Frontiers in human neuroscience*, 7.

Gombolay, M.C., Gutierrez, R.A., Clarke, S.G., Sturla, G.F. and Shah, J.A., 2015. Decision-making authority, team efficiency and human worker satisfaction in mixed human–robot teams. Autonomous Robots, 39(3), pp.293-312.

Kleinman, D., Baron, S., & Levison, W. (1971). A control theoretic approach to manned-vehicle systems analysis. *IEEE Transactions on Automatic Control*, 16(6), 824-832.

Levchuk, G., Pattipati, K., Fouse, A. and Serfaty, D., 2017. Application of free energy minimization to the design of adaptive multiagent teams. In *Disruptive Technologies in Sensors and Sensor System, SPIE DSO*.

Ljung, L., & Glad, T. (1994). *Modeling of dynamic systems*. Prentice Hall, Englewood Cliffs, N.J.

Pattipati, K. R., Kleinman, D. L., & Ephrath, A. R. (1983). A dynamic decision model of human task selection performance. *IEEE Transactions on Systems, Man, and Cybernetics*, (2), 145-166.

Pellerin, C., 2015. Work: Human-Machine Teaming Represents Defense Technology Future. *Department of Defense News*, accessed January 3, 2018, https://www.defense.gov/News/Article/Article/628154/work-human-machine-teaming-represents-defense-technology-future/

Perera, C., Zaslavsky, A., Christen, P. and Georgakopoulos, D., 2014. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys & Tutorials*, 16(1), pp.414-454.

Rivkin, J.W. and Siggelkow, N., 2003. Balancing search and stability: Interdependencies among elements of organizational design. *Management Science*, 49(3), pp.290-311.

Schläpfer, M., Bettencourt, L.M., Grauwin, S., Raschke, M., Claxton, R., Smoreda, Z., West, G.B. and Ratti, C., 2014. The scaling of human interactions with city size. *Journal of the Royal Society Interface,* 11(98), p.20130789.

Siggelkow, N. and Rivkin, J.W., 2005. Speed and search: Designing organizations for turbulence and complexity. *Organization Science*, 16(2), pp.101-122.

Smith, O. J. (1959). A controller to overcome dead time. *ISA J.*, 6, pp. 28-33.

West, G., 2017. *Scale: The Universal Laws of Growth, Innovation, Sustainability, and the Pace of Life in Organisms, Cities, Economies, and Companies*. Penguin.

Whitmore, A., Agarwal, A. and Da Xu, L., 2015. The Internet of Things—A survey of topics and trends. *Information Systems Frontiers*, 17(2), pp.261-274.

Yedidia, J.S., Freeman, W.T. and Weiss, Y., 2003. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8, pp.236-239.

Yedidia, J.S., Freeman, W.T. and Weiss, Y., 2005. Constructing free-energy approximations and generalized belief propagation algorithms. *IEEE Transactions on information theory*, 51(7), pp.2282-2312.

Viewing Machines as Teammates: A Qualitative Study

Joseph B. Lyons,[1] Sean Mahoney,[1] Kevin T. Wynne,[2] Mark A. Roebke[3]

[1]Wright-Patterson AFB, OH 45433; [2]University of Baltimore, Baltimore, MD 21201; [3]AF Inst. Techn., Wright-Patterson AFB, OH 45433
[1]{Joseph.lyons.6, us.af.mil, Sean.mahoney.5} @us.af.mil; [2]kwynne@ubalt.edu; [3]Mark.roebke.ctr@afit.edu

Abstract

The current paper discusses the concept of human-machine teaming and presents data from a qualitative study regarding the components of human-machine teaming. The dimensions of teammate likeness from a human-robot interaction perspective were reviewed. These dimensions formed the basis of a coding scheme used to analyze qualitative data. The data were taken from a survey among US workers. Participants (N = 605) were asked to: 1) identify an intelligent technology that they use on a regular basis, 2) classify the interaction with that technology as a teammate or a tool, and 3) report the reasons why they viewed the relationship as a teammate or what it would take for the relationship to be viewed as a teammate (if they reported viewing it as a tool). Results demonstrated good consistency with an emerging model of teammate likeness as discussed in the literature. Notable divergences were found for individuals who reported the technology as a tool versus as a teammate.

Background

Humans are surrounded by advanced technology on a regular basis. Technology is often a ubiquitous aspect of our daily routines and for some, interactions with technology may overshadow interpersonal interactions. As the boundaries blur between the frequencies of human versus technology interactions, researchers have begun to examine the topic of human-machine teaming (Chen & Barnes, 2014; Groom & Nass, 2007). Whether one is working side-by-side in a factory with a Baxter robot, driving alongside an autonomous car or taxi, walking past a Knightscope robot patrolling a parking lot, working with a bomb disposal robot in a military scenario, or trekking with a Ground Dog robot in the austere mountains of Afghanistan, humans are increasingly likely to interact with robotic systems. While in most of these extant interactions humans would likely characterize the robot as a "tool" versus a "teammate", teaming perceptions are increasingly warranted as technology advances in both capability and interactive capacity (Ososky, Schuster, Phillips, & Jentsch, 2013). Teaming with robots (versus teleoperation) is believed to be one of the real game changers with more advanced robotic systems. Yet little is known about human-machine teaming in the context of psychological perceptions.

Teaming with Technology

Groom and Nass (2007) outline several components of effective teamwork which include: shared goals, shared awareness (i.e., shared mental models), the desire for interdependence, motivation toward team versus individual objectives, action toward team objectives, and trust among team members. These factors are paramount to team effectiveness, and studies in the management domain have confirmed the importance of many of these team performance characteristics (Cohen & Bailey, 1997; De Jong, Dirks, & Gillespie, 2016; Kozlowski & Bell, 2003; Salas, Cooke, & Rosen, (2008). Yet, what of these factors can/should apply toward machine partners? Wynne and Lyons (in press) define *autonomous agent teammate-likeness* as "the extent to which a human operator perceives and identifies an autonomous, intelligent agent partner as a highly altruistic, benevolent, interdependent, emotive, communicative and synchronized agent teammate, rather than simply an instrumental tool" (p3; Wynne & Lyons; in press). The model posited by Wynne and Lyons is further outlined below. It should be noted that it is the combination of the dimensions below rather than a single dimension alone that is believed to influence teammate-likeness perceptions.

Perceived Agency.

Robotic systems that have greater decision authority and greater capabilities to execute that decision authority should influence teammate perceptions. By definition, a teammate is an autonomous entity that can contribute to the team's goals. Machine partners also, should be perceived as agentic entities. Effective agents should be able to observe the environment, process relevant goal-oriented information, and act on the environment (Chen & Barnes, 2014) – hence exemplifying agency. A lack of perceived agency should infer lack of autonomy, and increase perceptions that are tool-like versus teammate-like.

Perceived Benevolence

A core assumption of a teammate is that the teammate has your best interests in mind. Teammates support one another and provide back up where needed. The same should hold true of machine partners. Benevolence is a core trust antecedent (Mayer, Davis, & Schoorman, 1995) and it has been discussed as a key factor in driving human-robot trust (Lyons, 2013). Understanding the intent of a robotic system is a key ingredient to acceptance of the technology (Lasota & Shah, 2015). In an experimental study, Lasota and Shah (2015) found that robots made better teammates with participants in a joint manual task when the robots were aware of the human's next action – thus adding predictability into the robot's future intent. Further, robots that convey empathy (attributed intent) are liked more by participants (Leite et al., 2013). Thus, perceived benevolence from the technology should be an important factor in deciding whether the technology is a tool versus a teammate.

Perceived Task Interdependence

Mutual interdependence is a cornerstone of what it means to be part of a team. Interdependence presupposes some commonality in tasks and goals. When teaming with a machine partner, it is likely that the machine and human will work on separate aspects of the same task. If structured appropriately, the task will be divided into task components that are appropriate for the human and components that are appropriate for the machine to maximize the overall effectiveness of the human-machine team. In any case, interdependence with the machine will likely increase the perception of the technology as a partner versus as a tool.

Relationship Building

Imagine a world where teammates only discussed task-related information – what a boring relationship! True team members engage each other at a social level in addition to the task. Ososky and colleagues (2013) suggests that for humans to view robotic systems as partners, the interactive affordances need to move from one-sided information-centric transmissions to more naturalistic dialogue-based interactions. This will move the communication process from merely task-based to more relationship/team building-focused. Research by Hamacher, Bianchi-Berthouze, Pipe, and Eder (2016) shows that when interacting with robots, humans prefer robots that are expressive and warm over robots that are just focused on the task. These team-focused communications can signal loyalty and help build rapport among team members which are important team processes. As such, relationship-building communications will likely influence the perception of the technology as a teammate versus as a tool.

Communication Richness

Related to the above dimension, human-agent teams should be capable of rich dialogue to convey task and team-based information between each other (Chen & Barnes, 2014). Rich communicative and social cues affordances may make robots more effective when interacting with humans (Mutlu, 2011). The key distinction between this dimension and the above dimension is that the above dimension discusses non-task-oriented communications which are geared toward team-building. The current dimension focuses on the richness of communication in general, which could include both task-oriented and non-task communications. Media richness is believed to facilitate team effectiveness due to the added social and task-based information that rich media can convey (Hanumantharao & Grabowski, 2006). The greater the richness of communication affordances between the human and the technology the greater the likelihood of the human viewing the technology as a teammate versus a tool.

Synchrony

Effective teams are comprised of team members who have a shared awareness of the task, the team, and the context. Indeed, shared awareness and more specifically, having synchronized mental models has been shown to enhance team effectiveness (Hinds & Mortensen, 2005). Shared mental models have also been hypothesized to be important for human-machine teams (Ososky et al., 2013). Having synchrony between team members allows the team to share a common perception of the team and its capabilities/limitations, the context – which facilitates joint adaptation, and the task – which enables the team members to anticipate the actions of others.

In summary, the current paper examines the components of human-machine teaming using the Teammate Likeness Model (Wynne & Lyons, in press) as a rubric. It was expected that perceptions of agency, benevolence, interdependence, relationship-building, communication richness, and synchrony would be associated with more teammate (versus tool) perceptions.

Method

Participants

Six hundred and five US workers responded to an open call for participation on Amazon's Mechanical Turk (MTurk). The MTurk workers were employed at least part-time and were above the age of 18. No other demographics were collected in this study. Participants were compensated for their participation.

Study Description and Items

As part of a larger study focused on trust in automated technologies, participants were asked to identify one "intelligent technology" that they use on a regular basis. The following definition and description was provided to participants:

> Intelligent technologies or autonomous systems are technologies that can decide how and when to interact with you during tasks, communicate and/or dialogue with you, and or technologies that can help you accomplish your goals. Examples might include things like autonomous cars, service robots, industrial robots, robotic assistants, navigation aids, Amazon Echo/Google Home, the Nest, Siri, etc.

Once a technology was identified, participants were asked to characterize the relationship they had with the technology as a teammate- or tool-like relationship. Next they were asked to discuss why they characterized the relationship as a teammate relationship or (if they earlier noted that the relationship was more tool-like) what it would take for the relationship to be viewed as a teammate. Thus, in either characterization, the present study sought to understand the components of human-machine teaming. These open-ended responses were, in turn, coded according to the scheme described below.

Coding Method

Four independent coders coded the open-ended item. Two raters coded the entire set and two others coded a portion of the data. All raters were first trained on the coding process and the dimensions. Next, all four raters coded the first 70 participants and the coding team met to discuss their ratings. The next 30 participants were coded together as a team and consensus coding was used for the first set of participants (100 in total). Two raters coded the remaining 505 responses. Approximately 5% of the data was not usable due to the participants saying things like “there is no way a machine can be a teammate”. The items were coded for the following dimensions. First, items were coded as either teammate or tool according to the participants. There was 100% agreement among the raters on this dimension. Next, the open-ended item was coded using the teammate likeness model as a guiding rubric. In addition to the six dimensions noted above, a seventh category of “humanness” was added based on initial coding training and consensus coding which noted a high frequency of responses like, “it should act like a human” or “it should be like a human”, or “it should have human qualities”. The two raters averaged over 90% agreement across the 7 dimensions. It was possible, and very common, for multiple dimensions to be coded in the same participant response. Example excerpts for each dimension are below.

> “For it to be more of a teammate it would have to do things without me asking” (Agency)

> “It is a teammate to me. We work together to achieve this great balance where I trust the technology and Nest achieves its goals in making sure my home's temperature is just the way I like it” (Benevolence)

> “I think of the nest as more of a teammate. This is because it is working with me to help me reach a goal of becoming more energy efficient and helping me save money by making changes together to make it work the best” (Interdependence)

> “I see this completely as a tool. It's a functional item outside of me. A teammate would need to be more human, more personal, and more emotionally connected. I don't feel any emotional connection but rather a fully tool-like use” (Relationship Building)

> “I think for it to become more of a teammate, it would have to do far more than it does already. While it seems very personable, I know what kind of 'data' it can tell me, and as far as that goes, it's not too personal. To be more of a teammate, it could recommend places to eat out of nowhere, randomly talk to me about things, chime in on conversations and just generally exhibit more human-like behavior” (Communication Richness)

> “…To be a teammate it would need to "understand" better. That is, instead of me anticipating it, it would need to anticipate my needs. It's smart, but still lacking sometimes.” (Synchrony)

> “I consider it more of a tool. It would have to become either an actual human being, or an AI that was so human like you couldn't tell it wasn't, in order for me to consider it a teammate” (Humanness)

Results

Four hundred and nine participants (68%) reported that they believed their technology was tool-like versus teammate-like. The remaining 32% reported the relationship as more teammate-like. Forty-one percent were home technologies, 31% mobile technologies, 15% navigation aids, 3% automotive, 3% robotic systems, and 7% were classified as “other”. The technologies were further broken down based on brands. Twenty-two percent were Amazon products (Alexa, Echo), 15% were Apple products (Siri, iPhone), 11% were Google Maps, 6% were Androids or Google Assist, 5% were Google Home, 3% were Nest, and less than 1% for each Tesla and iRobot. An additional 36% was classified as “other”. These classifications and brands were categorized a priori.

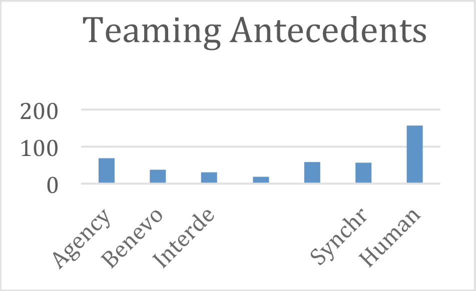


Figure 1. Frequency of Teaming Dimensions

As shown in Figure 1, Humanness was the most common dimension followed by Agency. However, each of the dimensions were noted by at least some participants. Relationship building was the least-noted teaming dimension.

Given the imbalance between participants who noted a teaming relationship and a tool-like perception, rather than report the absolute frequency of the dimensions by teaming versus tool perception, we reported the percentage of the dimension reported by the participants. In other words, the percentage of the responses for that dimension were calculated for each of those who viewed the technology as a teammate and those who viewed the technology as a tool.

As shown in Figure 2, those who viewed the technology as a teammate reported a greater percentage of benevolence and interdependence comments relative to those who viewed the technology as a tool. In contrast, when participants viewed the technology as a tool, they reported a greater percentage of comments that it would take humanness and communication richness to view the technology as a teammate.

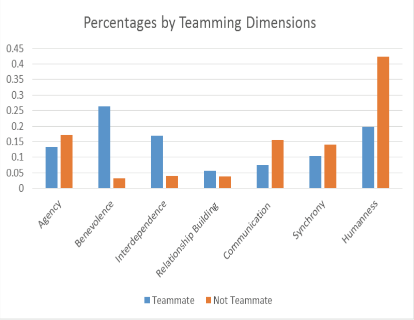


Figure 2. Relative Percentage of Teaming Dimensions

Discussion

The current study examined the construct of human-machine teaming using a qualitative sample and a broad cross-section of US workers. Participants were simply asked to list an intelligent technology that they use on a regular basis. The majority of these technologies were home use technologies such as the Amazon Echo or mobile technologies such as an iPhone equipped with Siri. Next, they were asked whether they viewed the technology as a tool or a teammate and why. An emerging model of teammate likeness was used to create a coding scheme for examining the qualitative data. The data largely confirm the presence of teammate perceptions of contemporary technologies and the results demonstrate the utility of the teammate-likeness construct overall.

The six dimensions of the teammate-likeness model were all invoked in the explanation for why the technology was (or could be) viewed as a teammate. Among the a priori set of dimensions, agency was most common followed by communication richness and synchrony. This shows the importance of viewing the technology as possessing some level of decision authority for a human to view the technology as a teammate. Furthermore, the findings for interdependence correspond to the management literature in terms of the importance of interdependence among individuals on a team. These are clearly important features for humans when considering the teaming relationship with technology. While the current results are interesting from a research perspective, care should be taken so that decision authority and interdependence with automated systems is done only when it has been carefully considered along with the potential limitations of such technologies. Human-machine teams may prove to more effective than either humans or technology alone, however great care must be taken in the design and implementation of technology in the workplace to avoid overreliance on reliable technology, as such overreliance can result in negative consequences if and when the technology (or the human) makes mistakes (see Onnasch, Wickens, Li, & Manzey, 2014 for a review).

Relationship building was the least common explanation provided by participants. It is possible that the affordances provided by the existing technologies did not allow for relationship building. While many of the technologies listed offer interactive features, many of these technologies lack the capacity to develop relationships. This dimension may be more relevant for future, more advanced technologies.

Interestingly, an unexpected dimension, humanness, was the most common response. The mere notion of a teammate may invoke anthropomorphic perceptions – similar to the more traditional human partner. What is unclear is whether or not the humanness dimension was a conglomeration of the other dimensions noted in the Teammate Likeness Model (Wynne & Lyons, in press). For instance, having agency, intent, rich communication affordances, and relationship-oriented may be facets of what people believe "humanness" consists of. However, it was impossible to test this speculation with the current data given that many participants simply said the technology should be "like a human" without saying what that actually means. Future research should examine the dimensions of human-machine teaming to determine if humanness is unique from the other components of teammate likeness.

A second interesting finding within the data is that participants noted different dimensions of teaming depending on whether they viewed the technology as a teammate versus as a tool. For participants who perceived the technology as a teammate, they reported a higher percentage of comments for benevolence and interdependence. For these individuals, the technology offered support and was believed to work interdependently with the humans. These factors are consistent with dimensions of team processes found in the literature on interpersonal teams (Cohen & Bailey, 1997; De Jong, Dirks, & Gillespie, 2016; Kozlowski & Bell, 2003). Human-machine teams appear to involve some of the similar team process variables. In contrast, when individuals viewed the technology as a tool, they believed that added communication richness and humanness would facilitate future teammate perceptions. It is interesting to note that these dimensions are what people might look for in prospective teaming relationships versus what they currently experience within teaming relationships.

Another notable finding in the current study is the fact that over 30% of the sample reported viewing the relationship with the technology as a teammate-based partnership. This suggests that human-machine teaming is a viable and fruitful topic of inquiry within the human factors and robotics literatures as individuals do establish very intimate connections with technologies. Future research is needed to validate the dimensions of human-machine teaming to better understand why and how humans make these connections with advanced technology.

References

Chen, J.Y.C.; & Barnes, M.J. 2014. Human-agent teaming for multirobot control: A review of the human factors issues. *IEEE Transactions on Human-Machine Systems*. 13-29.

Cohen, S. G.; & Bailey, D. E. 1997. What makes teams work: group effectiveness research from the shop floor to the executive suite, *Journal of Management.* 23: 239-290.

De Jong, B.A.; Dirks, K.T.; & Gillespie, N. 2016. Trust and team performance: A meta-analysis of main effects, moderators, and covariates. *Journal of Applied Psychology.* 101: 1134-1150.

Dzindolet, M.T.; Peterson, S.A.; Pomranky, R.A.; Pierce, L.G.; and Beck, H.P. 2003. The role of trust in automation reliance. *International Journal of Human-Computer Studies.* 58: 697-718.

Groom, V.; & Nass, C. 2007. Can robots be teammates? Benchmarks in human-robot teams. *Interaction Studies,* 8: 483-500.

Hamacher, A.; Bianchi-Berthouze, N.; Pipe, A.G.; & Eder, K. 2016. August. Believing in BERT: Using expressive communication to enhance trust and counteract operational error in physical human-robot interaction. *Proceedings of IEEE International Symposium on Robot and Human Interaction Communication (RO-MAN).* New York: IEEE.

Hanumantharao, S.; Grabowski, M. 2006. Effects of Introducing Collaborative Technology on Communications in a Distributed Safety-Critical System. *International Journal of Human-Computer Studies* 64: 714–726.

Hinds, P.J.; & Mortensen, M. 2005. Understanding Conflict in Geographically Distributed Teams: The Moderating Effects of Shared Identity, Shared Context, and Spontaneous Communication. *Organization Science* 16: 290–307.

Kozlowski, S.W.J.; & Bell, B.S. 2003. Work groups and teams in organizations. In W. Borman and D. Illgen (Eds.), *Handbook of psychology: Industrial and organizational psychology*, (Vol. 12, pp. 333-375). New York, NY: John Wiley & Sons Inc.

Lasota, P.A.; & Shah, J.A. 2015. Analyzing the effects of human-aware motion planning on close-proximity human-robot collaboration. *Human Factors* 57: 21-33.

Leite, I.; Pereira, A.; Mascarenhas, S.; Martinho, C.; Prada, R.; & Paiva, A. 2013. The influence of empathy in human-robot relations. *International Journal of Human-Computer Studies* 71: 250-260.

Lyons, J.B. 2013. Being transparent about transparency: A model for human-robot interaction. In D. Sofge, G.J. Kruijff, & W.F. Lawless (Eds.) Trust and Autonomous Systems: Papers from the AAAI Spring Symposium (Technical Report SS-13-07). Menlo Park, CA: AAAI Press.

Mayer, R.C.; Davis, J.H.; & Schoorman, F.D. 1995. An integrated model of organizational trust. *Academy of Management Review,* 20: 709-734.

Mutlu, B. 2011. Designing embodied cues for dialogue with robots. *AI Magazine*, 17-30.

Onnasch, L.; Wickens, C.D.; Li, H.; & Manzey, D. 2014. Human performance consequences of stages and levels of automation: An integrated meta-analysis. *Human Factors,* 56: 476-488.

Ososky, S.; Schuster, D.; Phillips, E.; & Jentsch, F. 2013. Building appropriate trust in human-robot teams. *Proceedings of AAAI Spring Symposium on Trust in Autonomous Systems* (pp. 60-65). Palo Alto, CA: AAAI.

Salas, E.; Cooke, N.J.; & Rosen, M.A. 2008. On teams, teamwork, and team performance: Discoveries and developments. *Human Factors,* 50: 540-547.

Wynne, K.T.; & Lyons, J.B. in press. An integrative model of autonomous agent teammate likeness. *Theoretical Issues in Ergonomics Science.*

Valuable Information and the Internet of Things

Ira S. Moskowitz,[1] **Stephen Russell**[2]
[1]Information Management and Decision Architectures Branch, Code 5580
Naval Research Laboratory, Washington, DC 20375
&
[2]Battlefield Information Processing Branch
Army Research Laboratory
Adelphi, MD 20783

Abstract

We investigate a theory for Value of Information (VoI) with respect to the Internet of Things (IoT) and IoT's intrinsic Artificial Intelligence (AI). In an environment of ubiquitous computing and information, information's value takes on a new dimension. Moreover, when the system in which such a volume of information exists is itself intelligent, the ability to elicit value, in context, will be more complicated. Classical economic theory describes the relationship between value and volume which, though moderated by demand, is highly correlated. In an environment where information is plentiful such as the IoT, the intrinsic intelligence in the system will be a dominant moderator of demand (e.g. self-adapting, self-operating, and self-protecting; controlling access). We examine Howard's (1966) VoI theory from this perspective and illustrate mathematically that Howard's focus on maximizing value obfuscates another important dimension, the guarantee of value.

Introduction

Shannon (Shannon 1948) laid the groundwork for information theory in his seminal work. However, Shannon's theory is a quantitative theory, not a qualitative theory. Shannon's theory tells you how much "stuff" you are dealing with, but it does not care if it is a cookie recipe or the plans for a time machine. The quality of "stuff" is irrelevant to Shannon theory. This is in contrast to Value of Information (VoI) theory, where we care about what, not necessarily how much, "stuff", we are considering. That is, Shannon is a purely quantitative theory, whereas any theory of information value must include a qualitative aspect that is equal in relevance as any quantitative measures.

This qualitative characteristic finds it way into many information-centric areas, particularly when humans or Artificial Intelligence is involved in decision making processes. For example, in (Russell, Moskowitz, and Raglin 2017) the authors, not surprisingly, state "We note that a purely quantitative approach to information is far from satisfactory." They then back this statement with discussions on Paul Revere, the Small Message Criteria (Moskowitz and Kang 1994), and steganography. This also discuss how Allwein (Allwein 2004) merged the work of Barwise and Seligman (Barwise and Seligman 1997) to Shannon's theory using the tools of channel theory from logic. However, these types of approaches do not offer immediate help to us with pragmatic issues that exist in the Internet of Things (IoT) where information is excessively plentiful.

The nature of the IoT is one of pervasive information, continuously gathered and acted on by fully or semi-autonomous devices and system. This notion creates an interesting paradox in the context of VoI. If the IoT ushers in unimaginable volumes of information, shouldn't the "value" of information decrease? Perhaps in the broader sense, e.g. *all* information's overall value may decrease, but certain information would still retain a value higher than most. This calls into question how applicable existing VoI theory would be in the context of the IoT and related decision-making. Moreover, the IoT itself is imbued with its own Artificial Intelligence, that manifests as self-star (self-*) behaviors. Self-* behaviors are (Babaoglu et al. 2005) autonomic behaviors (such as self-management, self-awareness, self-protecting, etc.) that imbue a device or system with an ability and understanding of its contribution (or value) to greater or external objectives/goals. The concept of the IoT's Artificial Intelligence (AI) brings additional constraints to understanding VoI, given such a pervasive information system. Like the limitations of Shannon's information theory (Shannon 1956), these considerations also create a fundamental issue of a solely quantitative theory of information's applicability to IoT decision-making.

We attempt to address this issue by examining a *Value of Information (VoI) theory* in the context of the IoT. Our thinking is heavily influenced by (Ponssard 1975) and especially by (Howard 1966). These works discuss how VoI is part of Decision Analysis. We attempt to make an optimal decision, based upon expected utility/value. Howard (Howard 1966) discusses how a company decides how much to bid on a contract based upon the *a priori* information available. In this situation the company attempts to maximize its expected profit. We note though that we disagree with how Howard obtained his "clairvoyant" results in the situation when additional information is available to the decision maker. Artificial Intelligence plays a major role in any consideration of the VoI because techniques, such as Machine Learning, can distill additional information from the IoT which can be used by a decision-maker.

IoT and AI

The Internet of Things is touted as the next wave in the era of computing (Gubbi et al. 2013) and has quickly been labeled the Internet of Everything (Roy and Chowdhury 2017). While the definition of the IoT may take many forms, there is little debate about the amount of information it will make available (Barnaghi, Sheth, and Henson 2013; Papadokostaki et al. 2017; Taherkordi, Eliassen, and Horn 2017) for decision related activties.

Quoting from (Moskowitz, Russell, and Jalian 2018)

> The Internet of Things (IoT) is the realization of interconnected and ubiquitous computing, pervasive sensing, and autonomous systems that can affect the physical world. ... The "things" that exist in the IoT can be generally thought of as physical or computational objects that label, sense, communicate, process, or actuate thereby bridging the physical and virtual worlds (Oriwoh and Conrad 2015; Pande and Padwalkar 2014). While there is no universally accepted definition of the IoT, the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) defines the IoT as "a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things..."

In (Moskowitz, Russell, and Jalian 2018), beyond providing a definition of the IoT, the authors showed how side channels in the IoT architecture can cause information to be covertly/steganographically transmitted from one place in the IoT to another. They argue that IoT will make so much information available, that new threats will emerge that are hiding in (information's) plain sight. We posit that the amount of available information in the IoT will change the supply and demand dynamic, resulting in a need for a new understanding of information's value. This relationship will likely follow an econometric view of value, where scarcity increases perceived and/or real value (Rymaszewska, Helo, and Gunasekaran 2017; Hansen and Serin 1997; Worchel 1992). What makes the IoT such an interesting arena for VoI research is that even where the number of bits is the same everywhere in the IoT, the value of those bits can differ upon where and when you are in a certain location in the IoT. For example, if my smart refrigerator sends a message that I only have one egg left (extrapolating from (Borgonovo 2017)), that information is only valuable to my cook, and it depends upon what s/he is preparing before s/he goes to the market again. Since I do not cook, that information is of no value to me. However, if my alarm system sends a message to my smart phone that there is someone in my house when no one is supposed to be home, that information may be of some value to my cook, but it is extremely valuable information to me.

The IoT changes one's normal perspective on how valuable information is obtained. We have many, many sources potentially sending information to a decision maker. This can be both good and bad. It can be good in that it enables us to reduce the uncertainty of some random variables. That is, one may be able to replace a continuous random variable with a large region of support, ideally with a Dirac delta distribution where we exactly know the information. That would be the ideal case, and is discussed in the later sections of Howard (Howard 1966). However, in a following section here, we will illustrate some mathematical differences with what Howard did, and discuss our findings with regard to perfect information (clairvoyance), which are also different.

The IoT can also be bad when it comes to the varied sources of potentially valuable decision-relevant information. Since the IoT is a huge conglomeration of processing and sensing devices, it is possible, and perhaps even likely, that contradictory information is obtained. Furthermore, the IoT will also be artificially intelligent itself (Etzion 2015; Elvy 2017). Machine learning algorithms are currently employed in the IoT at the local device and global usage levels (Ren and Gu 2015). Much of the machine learning approaches are implemented to provide the IoT with decision-making autonomy. In the next dimension of system intelligence, the IoT already is incorporating technologies to add increasing autonomic or self-star (self-*) behaviors. Self-* behaviors are those characteristics that form self-awareness and include self-organization, self-adaptation, and self-protection. The dependence on AI in IoT, in this context, is apparent. However the implications for AI enabled self-* behaviors to impact information value are less clear. Nonetheless, there is ample documentation in the literature about how AI can and will be employed as a gatekeeper for information (Camerer 2017; Conitzer et al. 2017), (Naseem and Ahmed 2017). It is through this merged lens of IoT and AI that we examine a theory of value of information. To provide grounding, we start with the work of Howard.

Reworking Howard's Initial Example

Howard's work (Howard 1966) takes a business approach to defining information value. In this section we borrow freely from Howard. We do not quote phrases for the sake of readability. We do not make any claims to this work, it re-works Howard's; the only novel thing in this section is our choice of notation and exposition.

This is a very practical problem of how much our company should bid to win a contract. If the bid is too high, it loses the contract. If the bid is too low, it gets the contract, but loses money on the deal. Therefore, our company attempts to place the bid that will get it the contract whilst maximizing its profit. The *information* that our company decides its bid upon is therefore of extreme importance and is considered to make up the sample space in question.

We assign a random variable $\mathcal{C}$ to be the cost of performing on the contract. Unfortunately, this cost is a probabilistic guess. We let the random variable $\mathcal{L}$ be the random variable representing the lowest bid of the competitors. Our company's bid is given by the random variable $\mathcal{B}$. Our company's profit is the random variable $\mathcal{V}$.

If $b > l$, our company loses the contract, and our profit is 0. If $b < l$ our company wins the contract and performs the work at a cost of c. Therefore the profit is $v = b - c$. Hence, similarly to (Howard 1966, Eq. 3) our company gets the contract in case of a tie ($b = l$). In terms of the random variables

$$\mathcal{V} = \begin{cases} \mathcal{B} - \mathcal{C}, & \text{if } \mathcal{B} \le \mathcal{L} \\ 0, & \text{if } \mathcal{B} > \mathcal{L} \;. \end{cases} \tag{1}$$

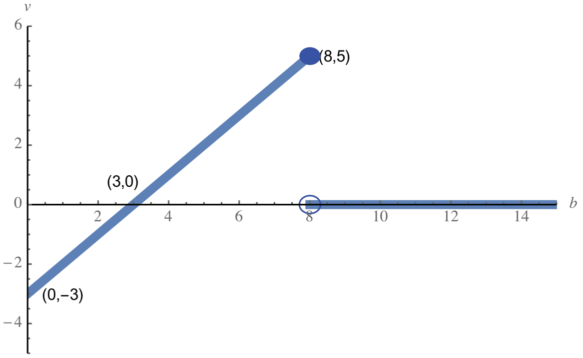


Figure 1: Profit $\mathcal{V} = v$, for $c = 3, l = 8, \mathcal{B} = b$

In Figure 1 we see the plot of Eq. (1) when $c = 3, l = 8$. There is no upper bound on what b may be, but v is always 0 for large enough b. Let us consider the density functions following (Howard 1966) but with a modified[1] notation of (Ross 1976). We have

$$f(v|b) = \iint\limits_{\mathbb{R}^2} f(v|b,c,l) \cdot f(c,l|b)\, dc\, dl \;. \tag{2}$$

This only makes sense when $0 \le b$. Our company never bids a negative amount, so any event involving $b < 0$ has zero probability, and conditional probability is thus not defined in that range.

We are interested in the expected value of profit conditioned on our bid. That is, we wish to determine $E(\mathcal{V}|b)$.

$$\begin{aligned} E(\mathcal{V}|b) &= \int_{-\infty}^{\infty} v \cdot f(v|b)\, dv \\ &= \iiint\limits_{\mathbb{R}^3} v \cdot f(v|b,c,l) \cdot f(c,l|b)\, dc\, dl\, dv \\ &= \iint\limits_{\mathbb{R}^2} f(c,l|b) \left(\int_{-\infty}^{\infty} v \cdot f(v|b,c,l)\, dv \right) dc\, dl \\ &= \iint\limits_{\mathbb{R}^2} E(\mathcal{V}|b,c,l) \cdot f(c,l|b) dc\, dl \end{aligned} \tag{3}$$

Now Howard makes two assumptions (Howard 1966, Eqs. 6,7) to simplify the problem.

Assumption 1. *The joint distribution of cost and lowest bid $\mathcal{C}, \mathcal{L}$ is independent of our company's bid $\mathcal{B}$. That is*

$$f(c,l|b) = f(c,l) \;.$$

Assumption 2. *Our company's cost $\mathcal{C}$ is independent of the lowest bid $\mathcal{L}$. That is*

$$f(c,l) = f(c)f(l) \;.$$

We realize that one could certainly argue the reality of these assumptions in all cases. Using Assumptions 1&2, we now have that

$$E(\mathcal{V}|b) = \iint\limits_{\mathbb{R}^2} E(\mathcal{V}|b,c,l) \cdot f(c)f(l) dc\, dl \;. \tag{4}$$

From Eq. (1) we see that once we set the values of $\mathcal{B}, \mathcal{C}, \mathcal{L}$ at b, c, l respectively, the density function of $\mathcal{V}$ becomes deterministic. That is

Theorem 1.

$$f(v|b,c,l) = \begin{cases} \delta(v - (b - c)), & \text{if } b \le l \\ \delta(v), & \text{if } b > l \;. \end{cases}$$

and therefore

$$E(\mathcal{V}|b,c,l) = \begin{cases} b - c, & \text{if } b \le l \\ 0, & \text{if } b > l \;. \end{cases}$$

Thus the following follows from Eq. (4)

Theorem 2. *Using Assumptions 1&2 we have*

$$\begin{aligned} E(\mathcal{V}|b) &= \iint\limits_{\mathbb{R}^2} E(\mathcal{V}|b,c,l) \cdot f(c)f(l) dc\, dl \text{ (as above)} \\ &= \int_{-\infty}^{\infty} (b - c) \left(\int_{b}^{\infty} f(l) dl \right) f(c) dc \qquad (5) \\ &= P(\mathcal{L} > b) \cdot \int_{-\infty}^{\infty} (b - c) f(c) dc \qquad (6) \\ &= [b - E(\mathcal{C})] \cdot P(\mathcal{L} > b) \;. \qquad (7) \end{aligned}$$

The above corresponds to (Howard 1966, Eq. 10). So, after our above assumptions, to obtain $E(\mathcal{V}|b)$ we only need the distribution of $\mathcal{L}$ and $E(\mathcal{C})$. Howard (Howard 1966) models $\mathcal{C}$ as a uniform distribution on $[0, 1]$ which implies $E(\mathcal{C}) = \frac{1}{2}$.

We relax what Howard did, and model the distribution of $\mathcal{C}$ such that $E(\mathcal{C}) = \frac{1}{2}$. We also follow Howard and model $\mathcal{L}$ as a uniform distribution on $[0, 2]$.

Thus, we say that the *base Howard example* is $\mathcal{L} = U[0, 2]$ and $E(\mathcal{C}) = \frac{1}{2}$.

The above gives us $P(\mathcal{L} > b) = \frac{1}{2}(2 - b), b \le 2$ (0 for $b > 2$). Of course, we do not consider $b < 0$ as discussed earlier. Therefore, we arrive at

$$E(\mathcal{V}|b) = \frac{1}{2}(2 - b)\left(b - \frac{1}{2}\right), 0 \le b \le 2 \;. \tag{8}$$

So we see that $E(\mathcal{V}|b) = -\frac{1}{2}\left[b^2 - \frac{5}{2}b + 1\right]$ is a simple quadratic and that $\frac{d}{db}E(\mathcal{V}|b) = -b + \frac{5}{4}$, so $E(\mathcal{V}|b)$ obtains a maximum of 9/32 when $b = 5/4$.

[1] For typographical simplicity we do not include the sub-index of the density function when the context is clear. That is, for example, we write $f(x)$ instead of $f_X(x)$, however, the complete notation is taken as being understood.

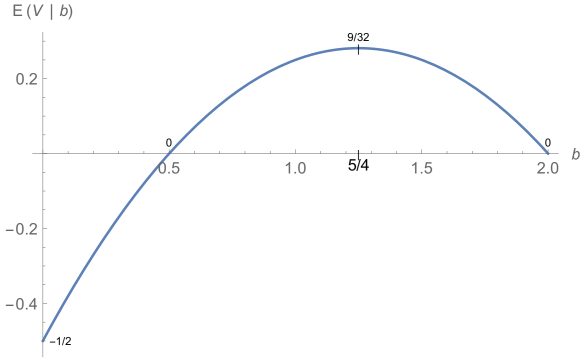


Figure 2: $E(\mathcal{V}|b)$

We define

$$\lceil\langle\mathcal{V}\rangle\rceil_b \triangleq \max_b E(\mathcal{V}|b) \ .$$

Therefore we see that when $E(\mathcal{C}) = .5$ and $\mathcal{L} = U[0,1]$,

$$\lceil\langle\mathcal{V}\rangle\rceil_b = 9/32 \ .$$

We are in agreement with everything that Howard has done to this point. What we do not agree with is how he used the concept of clairvoyance for additional information that may be learned. We note that the concept of clairvoyance is also discussed in (Borgonovo 2017, Ch. 11). We go back to this later in this paper.

Value Discussion

We see from the above that the expected value of a random variable is very important to a decision-maker. It is the value of the information that is used. This is important in the IoT because it will be the source of the information, moderated by the AI that either provides it, modifies it, or protects it. From this perspective the IoT may provide all of the information, too much information, or a limited amount of the information. We see in the above example, that we do not need the entire cost, given Howard's assumptions, only the mean of the cost. Therefore, it need not take that many bits of needed valuable information. Extending Howard's notion, what is the information we have so far and what is its value?

1. Equation 1: Modeling equation.
2. Equation 2: Standard probability theory.
3. Assumption 1: Independence of our company's bid.
4. Assumption 2: Cost and lowest bid independence.
5. Behavior of $\mathcal{C}$.
6. Behavior of $\mathcal{L}$.

Let us just concentrate on the last two items for now. What we have actually used so far is only the mean of $\mathcal{C}$, and for simplicity we set

$$\mu \triangleq E(\mathcal{C}).$$

The distribution of $\mathcal{L}$ is given by its density function $f_L(l)$. Modifying *this* information changes the quantity we care about, that is:

What is the "value" of the information in items 5 and 6 above in how it affects $\lceil\langle\mathcal{V}\rangle\rceil_b$? Does the shape of the graph change, does the maximum behavior change, etc?

We return to Eq. (4) to see the impact of changes of the information in items 5 and 6. First, let us change the distribution of $\mathcal{L}$ so it is uniformly distributed on $[0, L]$, $L > 0$, instead of $[0, 2]$.

We see that $P(\mathcal{L} > b) = \frac{1}{L}(L-b), b \leq L$ (0 for $b > L$). So, we see that in general for arbitrary positive μ we have

$$\begin{aligned} E(\mathcal{V}|b) &= \frac{1}{L}(L-b)\,(b-\mu)\,, 0 \leq b \leq L & (9)\\ &= -\frac{1}{L}\left(b^2 - [L+\mu]b + L\mu\right) \ . & (10)\end{aligned}$$

Simple calculus shows that the value b_o that maximizes $E(\mathcal{V}|b)$ is either the critical point $b_c = \frac{L+\mu}{2}$, if $b_c \leq L$, or the boundary point L if $\mu > L$. Thus,

$$\lceil\langle\mathcal{V}\rangle\rceil_b = \begin{cases} \frac{(L-\mu)^2}{4L}, \text{ with } b_o = \frac{L+\mu}{2}, & \text{if } 0 \leq \mu < L \\ 0, \text{ with } b_o = L, & \text{if } \mu \geq L \ . \end{cases} \tag{11}$$

We see that the only interesting case is when $0 < \mu < L$, which makes physical sense. We call this the non-trivial region, and denote the function defined on that region as $\langle\langle\mathcal{V}\rangle\rangle$.

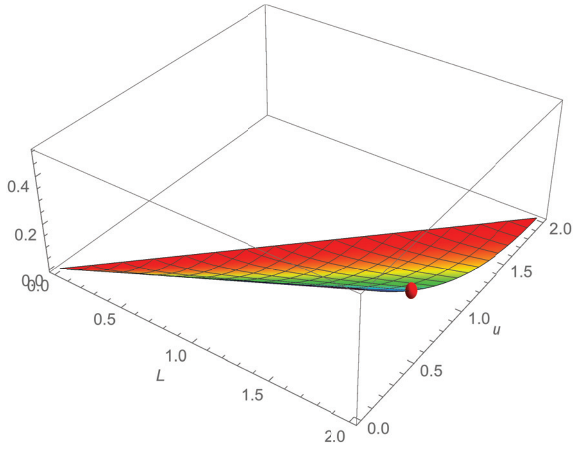


Figure 3: Surface plot of non-trivial values, for $L \in [0,2], \mu \in [0,2])$, of $\lceil\langle\mathcal{V}\rangle\rceil_b$, with point $(L = 2, \mu = .5, \lceil\langle\mathcal{V}\rangle\rceil_b = 9/32)$ highlighted.

Note that we also have

$$\frac{\partial\lceil\langle\mathcal{V}\rangle\rceil_b}{\partial L} = \begin{cases} \frac{1}{4}\left(1 - \left(\frac{\mu}{L}\right)^2\right) > 0, & \text{if } 0 \leq \mu < L \\ 0, & \text{if } \mu \geq L \ . \end{cases} \tag{12}$$

and

$$\frac{\partial\,\langle\langle\mathcal{V}\rangle\rangle}{\partial\mu} = \frac{1}{2}\left(\frac{\mu}{L} - 1\right) < 0 \ . \tag{13}$$

So, in the non-trivial region, increasing L increases $\lceil\langle\mathcal{V}\rangle\rceil_b$, and decreasing μ decreases $\lceil\langle\mathcal{V}\rangle\rceil_b$.

Let us pause and think about VoI. Is there any additional value in learning more about $\mathcal{C}$ other than its mean? No! This is an important understanding.

Also, if we are at a point in the non-trivial region, what is more important to learn about w.r.t. $\langle\langle\mathcal{V}\rangle\rangle$, a change in L or a change in $E(\mathcal{C})$? That is, if we have to prioritize the information that is sent to a decision-maker and we can only send one "fact" at a time which one would we send first, information about a change in L or $E(\mathcal{C})$? Consider the total differential

$$d\,\langle\langle\mathcal{V}\rangle\rangle = \frac{\partial\,\langle\langle\mathcal{V}\rangle\rangle}{\partial L}dL + \frac{\partial\,\langle\langle\mathcal{V}\rangle\rangle}{\partial \mu}d\mu \tag{14}$$

$$= \frac{1}{4}\left(1 - \left(\frac{\mu}{L}\right)^2\right)dL - \frac{1}{2}\left(1 - \frac{\mu}{L}\right)d\mu\,. \tag{15}$$

Thus, using $1 - x^2 = (1-x)(1+x)$, we see that

$$\left|\frac{\partial\,\langle\langle\mathcal{V}\rangle\rangle}{\partial L}\right| < \left|\frac{\partial\,\langle\langle\mathcal{V}\rangle\rangle}{\partial \mu}\right| < 2\left|\frac{\partial\,\langle\langle\mathcal{V}\rangle\rangle}{\partial L}\right|\,. \tag{16}$$

So, in the infinitesimal sense the value of $E(\mathcal{C})$ is more important than the value of L, but not by much. Therefore, if we have to prioritize information sent to a decision maker, it should be $E(\mathcal{C})$, then L.

Of course, all of the above is based upon the fact that we know the optimal $b_o = \frac{L+\mu}{2}$, which we learned from our above assumptions and calculations.

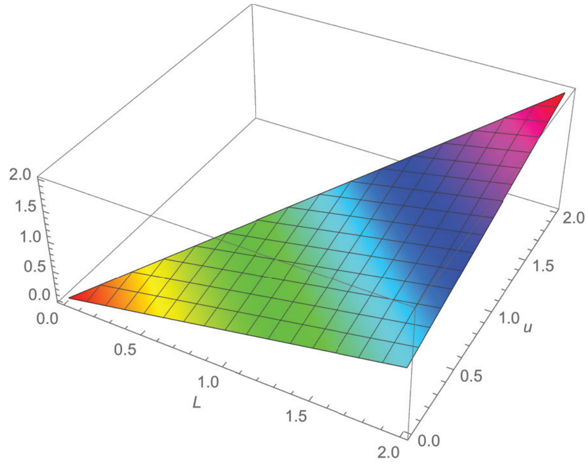


Figure 4: Surface plot of b_o in the non-trivial region for $L \in [0,2], \mu \in [0,2]$.

Generalization

Let us summarize the above in generality.

1. We are given distributions on $\mathcal{L}$ and $\mathcal{C}$.

2. $$\mathcal{V} = \begin{cases} \mathcal{B} - \mathcal{C}, & \text{if } \mathcal{B} < \mathcal{L} \\ 0, & \text{if } \mathcal{B} > \mathcal{L}\,. \end{cases}$$

3. $\mathcal{L}$ and $\mathcal{C}$ are independent of our company's bid $\mathcal{B}$.

4. Our company's cost $\mathcal{C}$ is independent of the lowest bid $\mathcal{L}$.

Thus,

$$E(\mathcal{V}|b) = [b - E(\mathcal{C})]\cdot P(\mathcal{L} > b) \;\text{ and now, in general,}$$

$$\lceil\langle\mathcal{V}\rangle\rceil_b \triangleq \max_b E(\mathcal{V}|b)\,.$$

Assuming that $\frac{d}{db}E(\mathcal{C}) = \frac{d}{db}f(l) = 0$ (which is not a far stretch from the statistical independence we have assumed of the underlying random variables), we have $\frac{d}{db}E(\mathcal{V}|b) = P(\mathcal{L} > b) - [b - E(\mathcal{C})]\cdot f_{\mathcal{L}}(b)$, where the term $f_{\mathcal{L}}(b)$ is the density function $f(l)$ of $\mathcal{L}$ evaluated at $l = b$. Thus, the optimal b_o in the non-trivial region solves the integral equation

$$b = E(\mathcal{C}) + \frac{P(\mathcal{L} > b)}{f_{\mathcal{L}}(b)} = E(\mathcal{C}) + \frac{1}{f_{\mathcal{L}}(b)}\int_b^\infty f_{\mathcal{L}}(l)dl\,,$$

$$\text{and in the non-trivial region } \lceil\langle\mathcal{V}\rangle\rceil_b = \frac{\Big(P(\mathcal{L} > b_o)\Big)^2}{f_{\mathcal{L}}(b_o)}\,.$$

Clairvoyance about $\mathcal{C}$

Let us go back to Eq. (4), but now let us assume that our company knows the cost $\mathcal{C}$. In this case our company will never bid less than the cost, or it will lose money! Note that our results in this section differ from Howard's results on clairvoyance.

Assumption 3. *Our company has knowledge of the cost.*

We must modify Thm. (1) so that

$$E(\mathcal{V}|b,c,l) = \begin{cases} b - c, & \text{if } c \le b \le l \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

So we have that

$$E(\mathcal{V}|b) = \iint\limits_{\mathbb{R}^2} E(\mathcal{V}|b,c,l)\cdot f(c)f(l)dc\,dl \text{ (as above)}$$

$$= \int_{-\infty}^{b}(b-c)\left(\int_b^\infty f(l)dl\right)f(c)dc \tag{18}$$

$$= P(\mathcal{L} > b)\cdot\int_{-\infty}^{b}(b-c)f(c)dc \tag{19}$$

$$= \left[b\cdot P(\mathcal{C} \le b) - \int_{-\infty}^{b} cf(c)dc\right]\cdot P(\mathcal{L} > b) \tag{20}$$

We will go through an example similar to what we did before. Previously, we followed Howard and modeled $\mathcal{C}$ so that $E(\mathcal{C}) = 1/2$, and $\mathcal{L} = U[0,2]$. Note that before the distribution of $\mathcal{C}$ did not matter, only its mean. We see from the above that this is no longer true. Let us try some examples.

Example 1: $\mathcal{L} = U[0,2]$, and $P(\mathcal{C} = 1/2) = 1$.
So we have that $f(c) = \delta(c - 1/2)$, and Eq. (20) becomes

$$E(\mathcal{V}|b) = \begin{cases} [b - 1/2]\cdot P(\mathcal{L} > b), & \text{if } 1/2 < b \le 2 \\ 0, & \text{otherwise.} \end{cases} \tag{21}$$

This simplifies to

$$E(\mathcal{V}|b) = \begin{cases} (b - 1/2)\left(\frac{2-b}{2}\right), & \text{if } 1/2 < b \le 2 \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

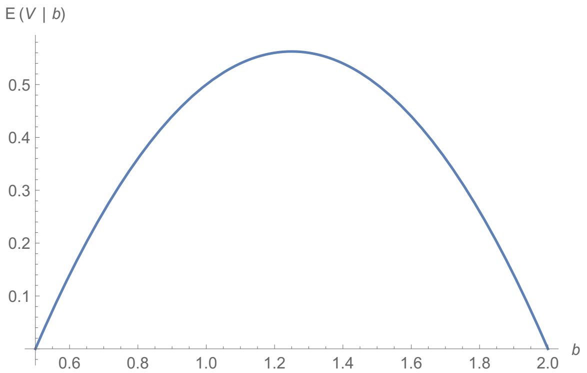


Figure 5: $E(\mathcal{V}|b) \geq 0$ when our company knows the cost, $\mathcal{L} = U[0,2]$, and $P(\mathcal{C} = 1/2) = 1$.

In Example 1, $E(\mathcal{V}|b)$ has a maximum value of 18/32 , when $b = 5/4$.

Example 2: $\mathcal{L} = U[0,2]$, and $\mathcal{C} = U[0,1]$.
Eq. (20) becomes

$$E(\mathcal{V}|b) = \begin{cases} \left[b \cdot \left(\frac{b-0}{1-0}\right) - \int_0^b c \cdot \frac{1}{1} dc\right] \cdot \left(\frac{2-b}{2}\right), & \text{if } 0 \leq b \leq 1 \\ \left[b \cdot P(\mathcal{C} \leq 1) - \int_{-\infty}^{1} c dc\right] \cdot \left(\frac{2-b}{2}\right), & \text{if } 1 < b \leq 2 \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

This simplifies to

$$E(\mathcal{V}|b) = \begin{cases} \frac{b^2}{4}(2-b), & b \in [0,1] \\ [b - E(\mathcal{C})] \cdot \left(\frac{2-b}{2}\right) = \frac{1}{2}(2-b)\left(b - \frac{1}{2}\right), & b \in (1,2] \\ 0, & \text{otherwise.} \end{cases} \tag{24}$$

We note with interest that $E(\mathcal{V}|b)$ is a (once) differentiable function on $[0,2]$.

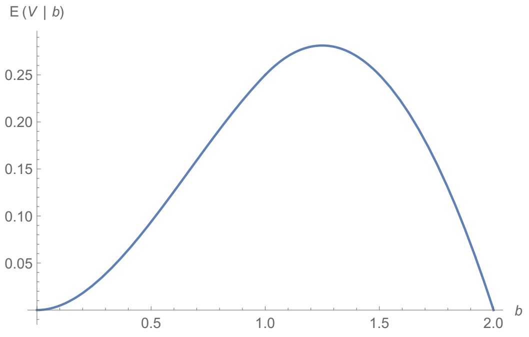


Figure 6: $E(\mathcal{V}|b) \geq 0$ when our company knows the cost, $\mathcal{L} = U[0,2]$, and $\mathcal{C} = U[0,1]$.

In Example 2, $E(\mathcal{V}|b)$ has a maximum value of 9/32 when $b = 5/4$, which is the same as Howard's base example.

We see that when our company knows the cost $\mathcal{C}$, the distribution, not just the mean, affects the behavior of $E(\mathcal{V}|b)$. We also see that knowledge of $\mathcal{C}$ guarantees that $E(\mathcal{V}|b) \geq 0$. That is, we never lose money.

Clairvoyance about $\mathcal{L}$

Now we are in the situation where our company knows the competitor's lowest bid, which is represented by $\mathcal{L}$. As before we assume that if our company's bid b ties with the competition's lowest bid l that our company wins the contract. Therefore, if we know l we bid l; this is done to win the contract *and* maximize profit. Note, if one finds this disturbing, we can always make the bid b a tiny amount less than l. Therefore, we see that $\mathcal{B}$ and $\mathcal{L}$ must be the same. Our company will always win the bidding, but it may lose money depending on the value of c. Therefore,

$$E(\mathcal{V}|b) = E(\mathcal{V}|l)\ . \tag{25}$$

Modifying Eq. (1), now differently than Eq. (17), we have

$$E(\mathcal{V}|b,c,l) = \begin{cases} l - c, & l \in \text{support of } \mathcal{L} \\ 0, & \text{otherwise.} \end{cases} \tag{26}$$

So we have that ($\mathcal{C}$ and $\mathcal{L}$ still independent):

$$\begin{align} E(\mathcal{V}|l) &= \int\limits_{\mathbb{R}} E(\mathcal{V}|c,l) \cdot f(c) dc \tag{27} \\ &= \int\limits_{\mathbb{R}} (l-c) \cdot f(c) dc \tag{28} \\ &= l \int\limits_{\mathbb{R}} f(c) dc - \int\limits_{\mathbb{R}} c \cdot f(c) dc \tag{29} \\ &= l - E(\mathcal{C}) \tag{30} \end{align}$$

when $l \in$ support of $\mathcal{L}$.

Example 3: $\mathcal{L} = U[0,L]$, and $E(\mathcal{C}) = 1/2$.
Below we show a plot of $E(\mathcal{V}|l) = l - .5$ against l for $\mathcal{L} = U[0,2]$ and $E(\mathcal{C}) = 1/2$.

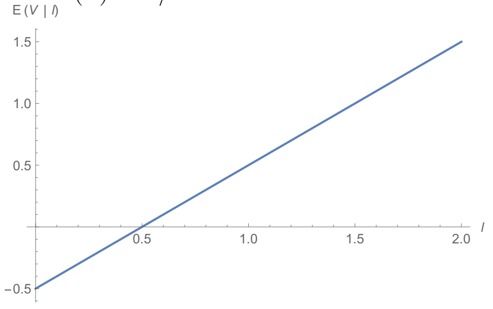


Figure 7: $E(\mathcal{V}|b)$such that $E(\mathcal{C}) = 1/2$, and our company knows the lowest bid, distributed as $\mathcal{L} = U[0,2]$.

Thus, for Example 3, $\lceil \langle \mathcal{V} \rangle \rceil_b = 1.5$, achieved when $b = 2$.

Note that $E(\mathcal{V}|b)$ is a linear function of $b = l$ and that it can be negative, zero (once), or positive depending on the support of $\mathcal{L}$. Furthermore the maximum of $E(\mathcal{V}|b)$ is achieved when b is the largest value of l in the support of $\mathcal{L}$. Heuristically, another way of saying this is that the maximum is achieved for the largest value of l such that $P(L \in (l - dx, l)) \neq 0$.

Unlike with the clairvoyant knowledge of $\mathcal{C}$, with knowledge of $\mathcal{L}$, $E(\mathcal{V}|b)$ may be negative, but the profit may be much larger. So, knowledge of $\mathcal{C}$ gives us non-negative profit, whereas knowledge of $\mathcal{L}$ gives us larger potential profit. This is in-line with what Howard obtained.

Clairvoyance about $\mathcal{C}$ and $\mathcal{L}$

Now we combine both pieces of information, the bid b will never be less than c, and it will always match l, therefore we must modify Eq. (1) again, now different than Eq. (25) because $\mathcal{L}$ is more restricted, resulting in

$$E(\mathcal{V}|b, c, l) = \begin{cases} l - c, & \text{if } c \leq l, \text{ and } l \in \text{support of } \mathcal{L} \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

So we have an assumption of independence between $\mathcal{C}$ and $\mathcal{L}$

$$E(\mathcal{V}|l) = \int\limits_{\mathbb{R}} E(\mathcal{V}|c, l) \cdot f(c) dc \tag{32}$$

$$= \int_{-\infty}^{l} (l - c) \cdot f(c) dc \tag{33}$$

$$= l \cdot P(\mathcal{C} < l) - \int_{-\infty}^{l} c \cdot f(c) dc \tag{34}$$

when $l \in$ support of $\mathcal{L}$.

Example 4: $\mathcal{L} = U[0, 2]$, and $\mathcal{C} = U[0, 1]$.

$$E(\mathcal{V}|b) = \begin{cases} l \cdot \int_0^l dc - \int_0^l c\, dc & \text{if } 0 \leq l < 1 \\ l \cdot \int_0^1 dc - \int_0^1 c\, dc, & \text{if } 1 \leq l \leq 2, \\ 0, & \text{otherwise. Thus,} \end{cases} \tag{35}$$

$$E(\mathcal{V}|b) = \begin{cases} \frac{l^2}{2}, & \text{if } 0 \leq l < 1 \\ l - .5, & \text{if } 1 \leq l \leq 2, \\ 0, & \text{otherwise.} \end{cases} \tag{36}$$

Below we plot $E(\mathcal{V}|b) = E(\mathcal{V}|l)$ against b for $\mathcal{L} = U[0, 2]$, and $\mathcal{C} = U[0, 1]$.

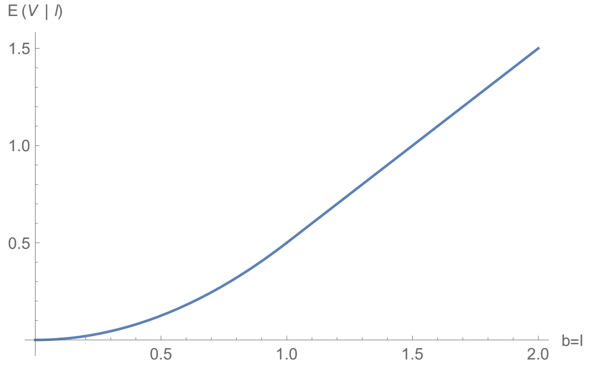


Figure 8: $E(\mathcal{V}|b)$ for when $\mathcal{C} = U[0, 1]$ and our company knows the lowest bid has distribution $\mathcal{L} = U[0, 2]$.

Thus, for Example 4, $\lceil \langle \mathcal{V} \rangle \rceil_b = 1.5$, achieved when $b = 2$. Note that the behavior of Ex. 3 and Ex. 4 are identical for $b > 1$. The difference is that if we know $\mathcal{C}$, we may never place a bid that will lose money.

Conclusion

The Internet of Things will provide a rich environment, supplying volumes of information for nearly every aspect of humans' activities and environments. The IoT will gain ever increasing amounts of Artificial Intelligence, that will only provide greater degrees of autonomic capabilities and self-star behaviors. This AI-enriched IoT environment will change the fundamental notions of information value for decision-making by producing huge quantities of information that are managed by the AI functionality. Like Shannon's information theories, our understanding of VoI theory will implicitly go beyond just a quantitative concept to include qualitative notions. However there is surprisingly little literature that examines VoI in the context of the IoT. In this paper we have extended Howard's (Howard 1966) VoI theory and examine a generalization of that notion towards a guarantee of a minimal value.

We presented a re-work of Howard's theoretical problem and solution identifying some limitations in his treatment of a random variable, relative to VoI. Howard's idea of *clairvoyance*, or insight into future information (and thus its value) treats the value of the random variable deterministically, rather than probabilistically. By giving the random variable a probabilistic context, such as would be the case of information provided by the AI-enabled IoT, the theoretical handling of clairvoyance changes. We see, as did Howard, that knowledge about $\mathcal{L}$ is more important than knowledge about $\mathcal{C}$ when it comes to maximizing $E(\mathcal{V}|b)$. But we show knowledge of $\mathcal{C}$ guarantees that we will never have a negative expected profit. Therefore, the value of information depends on what one is trying to do, or the contextual objective. This qualitative consideration must be kept in mind in further research on VoI.

We explained the relevance of our approach in this paper's section on IoT and AI, and we have taken the opportunity of adjusting Howard's seminal theory to provide an extended foundation for the Value of Information theory in the IoT. One must keep in mind that AI techniques, such as machine learning and artificial reasoning, when employed in the IoT for self-star system behaviors, will require additional consideration for managing information provided to a human or machine decision-maker. While we continued with Howard's "market" context in this paper for explainability and theoretic continuity, our future work will examine the implications of our theoretical VoI guarantee, described herein, in an IoT-specific experimental simulation or empirical study.

Acknowledgements

We acknowledge the assistance of Swmbo Heilizer. We also thank William F. Lawless for his careful reading of the draft version of this paper and his helpful suggestions.

References

Allwein, G. 2004. A qualitative framework for Shannon information theories. In *Proc. NSPW*, 23–31.

Babaoglu, O.; Jelasity, M.; Montresor, A.; Fetzer, C.; Leonardi, S.; van Moorsel, A.; and van Steen, M. 2005. The self-star vision. *Self-star properties in complex information systems* 397–397.

Barnaghi, P.; Sheth, A.; and Henson, C. 2013. From data to actionable knowledge: Big data challenges in the web of things [Guest Editors' Introduction]. *IEEE Intelligent Systems* 28(6):6–11.

Barwise, J., and Seligman, J. 1997. *Information Flow: The Logic of Distributed Systems*, volume 44 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press.

Borgonovo, E. 2017. *Sensitivity Analysis*. Number 251 in International Series in Operations Research & Management Science. Springer. chapter Value of Information —11, 93–100.

Camerer, C. F. 2017. Artificial Intelligence and Behavioral Economics. In *Economics of Artificial Intelligence*. University of Chicago Press.

Conitzer, V.; Sinnott-Armstrong, W.; Borg, J. S.; Deng, Y.; and Kramer, M. 2017. Moral Decision Making Frameworks for Artificial Intelligence. In *AAAI*, 4831–4835. 00011.

Elvy, S.-A. 2017. The Artificially Intelligent Internet of Things and Article 2 of the Uniform Commercial Code.

Etzion, O. 2015. When artificial intelligence meets the internet of things. In *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems*, 246–246. ACM.

Gubbi, J.; Buyya, R.; Marusic, S.; and Palaniswami, M. 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future generation computer systems* 29(7):1645–1660. 03114.

Hansen, P. A., and Serin, G. 1997. Will low technology products disappear?: The hidden innovation processes in low technology industries. *Technological Forecasting and Social Change* 55(2):179–191.

Howard, R. A. 1966. Information value theory. *IEEE Trans on Systems Science and Cybernetics* 2(1):22–26.

Moskowitz, I. S., and Kang, M. H. 1994. Covert channels — here to stay? In *Proc. COMPASS'94*, 235–243. Gaithersburg, MD: IEEE Press.

Moskowitz, I. S.; Russell, S.; and Jalian, B. 2018. Steganographic internet of things: Graph topology timing channels. to appear: Proc. AAAI workshop on Artificial Intelligence for Cyber Security (AICS).

Naseem, S., and Ahmed, K. 2017. Information Protection in Cognitive Science. *International Journal of Computer Science and Network Security (IJCSNS)* 17(3):1.

Oriwoh, E., and Conrad, M. 2015. 'Things' in the Internet of Things: Towards a definition. *International Journal of Internet of Things* 4(1):1–5.

Pande, P., and Padwalkar, A. R. 2014. Internet of Things–A Future of Internet: A Survey. *International Journal* 2(2).

Papadokostaki, K.; Mastorakis, G.; Panagiotakis, S.; Mavromoustakis, C. X.; Dobre, C.; and Batalla, J. M. 2017. Handling Big Data in the Era of Internet of Things (IoT). In *Advances in Mobile Cloud Computing and Big Data in the 5G Era*. Springer. 3–22.

Ponssard, J.-P. 1975. A note on information value theory for experiments defined in extensive form. *Management Science* 22(4):449–454.

Ren, F., and Gu, Y. 2015. Using Artificial Intelligence in the Internet of Things. *ZTECOMMUNICATIONS* 1.

Ross, S. 1976. *A First Course in Probability*. Macmillan.

Roy, S., and Chowdhury, C. 2017. Integration of Internet of Everything (IoE) with Cloud. In *Beyond the Internet of Things*. Springer. 199–222.

Russell, S.; Moskowitz, I. S.; and Raglin, A. 2017. *Human Information Interaction, Artificial Intelligence, and Errors*. Springer. 71–101.

Rymaszewska, A.; Helo, P.; and Gunasekaran, A. 2017. IoT powered servitization of manufacturing–an exploratory case study. *International Journal of Production Economics*.

Shannon, C. E. 1948. A Mathematical Theory of Communication. *Bell Systems Technical Journal* 27:379–423, 623–656.

Shannon, C. E. 1956. The bandwagon. *IRE Transactions on Information Theory* 2(1):3.

Taherkordi, A.; Eliassen, F.; and Horn, G. 2017. From IoT big data to IoT big services. In *Proceedings of the Symposium on Applied Computing*, 485–491. ACM.

Worchel, S. 1992. Beyond a commodity theory analysis of censorship: When abundance and personalism enhance scarcity effects. *Basic and Applied Social Psychology* 13(1):79–92.

AI Enabled Blockchain Smart Contracts: Cyber Resilient Energy Infrastructure and IoT

Michael Mylrea
Senior Manager | Cyber Security & Energy Technology |
Blockchain Energy Lead, Pacific Northwest National Laboratory
michaelmylrea@pnnl.gov

Abstract

The commoditization of trust has been the topic of science fiction, futuristic novels and theoretical study for the last century. Advances in blockchain and artificial intelligence technology continue to make science-fiction a reality, automating and replacing the need for 3rd party intermediaries and other trust mechanisms, potentially disrupting many critical industries. Blockchain enabled smart contracts show potential to exchange value without third party trust mechanisms. The combination of artificial intelligence, cryptography, distributed trust algorithms or smart contracts have paved the way to a more efficient and secure way to exchange value, goods and services. This paper explores how blockchain technology could potentiality automate and modernize energy and the internet of things to help evolve energy infrastructure to an increasingly automated, distributed, clean and resilient system. This is timely as the U.S. power grid and the array of things that it connects to is a complex system of systems in which the nation's economy, national security and livelihood depends on.

Introduction

Blockchain is defined as a distributed data base or digital ledger that records transactions of value using a cryptographic signature that is inherently resistant to modification (Tapscott 2016). Combining blockchain based smart contracts with machine learning algorithms presents an opportunity to increase the speed, scale, security and autonomy of complex, distributed internet of things (IoT) environments. Certainly, the need for third parties in executing a transaction will be reduced or even be replaced when an autonomous smart contract can execute and exchange value and services via an autonomous agent. But who will be held responsible when there is an error or when the contract is not successfully executed? While the data and exchange of value captured in blockchain might be immutable, or at least very hard to manipulate, what if the algorithm that establishes the terms of the contract executed is written by a AI agent. What additional challenges and potential solutions should be explored to via AI enabled blockchain solutions to distribute and automate IoT in a more secure way?

This paper explores these questions through an innovative blockchain smart contract application to electricity infrastructure and the array of networked things that are increasing connected responsible for energy generation, transmission, distribution and consumption. This use case highlights how AI enabled blockchain solutions may help increase cyber resilience and optimize complex exchanges of distributed energy resources by encrypting, monitoring and automating transactions and removing third parties. With billions of IoT devices sensing and exchanging information, AI enabled blockchain solutions could also help better analyze data sets from thousands of variables (industrial control system anomalies, frequency, load and voltage changes) and organize them into weighted relationships, which could be tracked through a next generation blockchain solution. As data patterns in these variables are better understood via machine learning computer-based neural networks, the smart blockchain contract could be updated to better secure and exchange energy data.

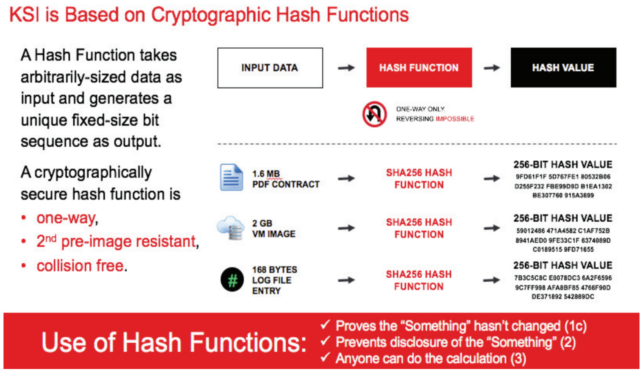
The AI Enabled Blockchain Opportunity: The Evolution of Public Key Infrastructure Encryption

Most cyber security solutions increase cost and reduce functionality in the name of integrity, confidentiality and availability. Blockchain solutions may prove to be an exception in that some applications can improve security and optimize the ability to exchange and track value. Indeed, some blockchain solutions add a layer of cryptography to track digital transactions, but many cyber security challenges remain for securing complex IoT environments. For one, IoT environments are often designed with functionality and cost in mind and security is often an afterthought. IoT often lacks encryption, basic patch management, uses default passwords and communicates in plain text. Poor source code, vulnerable design and improper configuration have also led to several major cyber incidents.

Another challenge with securing IoT from emerging cyber threats, is that public key infrastructure (PKI) solutions are often cost prohibitive and not scalable to realize the encryption requirements of IoT environments. Moreover, legacy systems and converged information technology (IT) and operational technology (OT) environments lack the necessary computer processing power to support some deployments of PKI. This is often seen with analogue equipment in substations and other critical infrastructures. Moreover, with PKI there is often a single authority that both issues and revokes the security certificate. If this authority is attacked and certificate is manipulated, all its users will potentially be vulnerable to cyber-attacks.

Thus, PKI must continue to evolve to secure IoT environments or a better solution needs to be scaled up. Blockchain keyless signature infrastructure KSI presents a potential path forward. KSI is a promising solution patented by Gaurdtime, one of the largest blockchain providers by revenue, which helps preserve the integrity of data exchanges and other digital transactions using a mathematical algorithm for authentication without the need for trusted keys or credentials. KSI authenticates IoT data at scale, in real time, providing immutable transaction data without several of the challenges of PKI. The following image further describes KSI's cryptographic hash function, highlight how the hash function can help prove the IoT device hasn't change, preventing the disclosure of sensitive IoT data and providing cryptographic proof that can be proven.

Researchers at Pacific Northwest National Laboratory, Guardtime Guardtime, the United States Department of Energy (DOE), Washington State University, Tennessee Valley Authority (TVA), Siemens and the Department of Defense Homeland Defense and Security Information Analysis Center (HDIAC) are developing a KSI enabled blockchain solution to help secure distributed energy IoT environments found in modern electricity infrastructure. This is especially important because as we modernize our energy infrastructure, the speed, size and complexity of energy data and transactions exchanged increases exponentially.


KSI is Based on Cryptographic Hash Functions

A Hash Function takes arbitrarily-sized data as input and generates a unique fixed-size bit sequence as output.

A cryptographically secure hash function is

- one-way,
- 2nd pre-image resistant,
- collision free.

Use of Hash Functions:

✓ Proves the "Something" hasn't changed (1c)
✓ Prevents disclosure of the "Something" (2)
✓ Anyone can do the calculation (3)

Figure 1. Gaurdtime's KSI blockchain is based on Cryptographic Hash Functions (Johnson 2017).

To help overcome these challenges, blockchain keyless signature infrastructure technology provides a unique value proposition in its potential to help optimize and secure these critical data sets from emerging cyber threats. AI enabled blockchain shows potential to enable critical energy delivery systems to be increasingly automated to respond to a naturally occurring weather event, cyber or cyber-physical hybrid attack, in a way that that some critical energy infrastructure functions become increasingly self-healing and resilient.

Blockchain's digital ledger and cryptography signed transaction data may help increase the trustworthiness and integrity energy transactions. Combined with machine learning and AI enabled energy delivery systems, these systems may also have more control and flexibility in automating, monitoring and auditing of complex energy exchanges at the grid's edge.

Combining AI and blockchain capability could also provide a real-time security response to unauthorized attempts to change critical EDS data, configurations, applications, and network appliance and sensor infrastructure. Autonomous detection of data anomalies and reduces burden with normalized evidence across a unified timeline for incident analysis. A data exchange platform using smart contracts for the automated trading and settlement of contracts in the electricity production value chain.

Distributed Consensus Algorithm

Blockchain is defined as a distributed data base or digitalledger that records transactions of value using a cryptographicsignature that is inherently resistant to modification (Tapscott 2016). Blockchain is a distributed database that maintains a continuously growing list of records, called blocks, secured from tampering and revision. Each block contains a timestamp and a link to a previous block. Blockchain-based smart contracts can be executed without human interaction (Franco 2014) and the data is more resistant to modification as the data in a block cannot be altered retroactively. Blockchain smart contracts are defined as technologies or applications that exchange value without intermediaries acting as arbiters of money and information (Tapscott 2016).

A keyless signature blockchain infrastructure (KSBI) differs from proof or work blockchain based crypto currencies as it is based around a concept of permission-based blockchain - to provide widely witnessed evidence on what can be considered the truth, independently of any single party and while retaining complete confidentiality of the original data. Another unique characteristic that differentiates the KSBI from other distributed ledger solutions are its ability to scale to industrial applications to add one trillion data items to the blockchain every second, and to verify the data item from the blockchain within the next second. The ability to transact data at sub second speeds is essential to handle the increasing data requirements of a modern power grid (Mylrea et al. 2017).

KSBI is based on Guardtime's patented technology keyless signature infrastructure (KSI_® which has been in production use since 2007, is employed by various world's governments – i.e. Estonia and Defense primes in United States - and is beginning to see adoption in the private sector for application for their systems and networks. A KSBI may also help realize several cybersecurity and compliance goals for the energy sector, such as:

Smart contracts: Smart contracts execute and record transaction in the blockchain load ledger through blockchain enabled advanced metering infrastructure (AMI). Blockchain based smart contracts may help facilitate consumer level exchange of excess generation from DER. This could provide additional storage and help substation load balancing from bulk energy systems. Moreover, smart contract data is secured in part through decentralized storage of all transactions of energy flows and business activities (Mylrea et al. 2017).

Secure Data Storage in Cryptographically Signed Distributed Ledger: Blockchain can help fill various optimization and security gaps and improve the state of the art in grid resilience by providing an atomically verifiable cryptographic signed distributed ledger to increase the trustworthiness, integrity and security of energy delivery systems at the edge. Blockchain can be used to verify time, user, transaction data and protect this data with an immutable crypto signed distributed ledger (Mylrea et al. 2017).

AI Enabled Blockchain Overview

Certainly, AI enabled blockchain will be disruptive and replace jobs, especially traditional 3rd parties that are replaced by new consensus algorithms and distributed trust mechanisms. Energy aggregators and meter readers could potentially be replaced by a dynamic distributed ledger. Blockchain innovation will also create new energy jobs, value, and markets. Even as technology empowers humans, it also changes the relationship between man and machine, technology and organizations, society and innovation. Autonomous blockchain organizations may distribute power and leadership via cryptographic votes that establish equity against a contract or even mission statement. For example, future energy organizations may have stakeholders govern what type of energy mix they would like and have that preference or willingness to pay be capture in a smart contract. Blockchain AI empowered energy organizations might be increasingly autonomous made up decentralized contractors and investors with power to vote, invest and delivery services based on an immutable smart contract that captures who, what, when and where services are executed and shared in a transparent immutable ledger.

The notion of a "self-bootstrapped" organizations with crypto equities leveraging independently contractors guided by decentralized blockchain voting has been explored (Levine 2014). Bit congress has established a blockchain based voting system. The country of Georgia is leveraging blockchain to facilitate real estate licensing. Estonia has established a privacy preserving secure virtual government using keyless signature infrastructure blockchain. These examples highlight how technology can help distribute trust and reduce redundancy in everything from billing to middle management, creating new value for organizations in an increasingly decentralized autonomous society. Reducing redundancy creates new value and more competitive organizations (Lawless 2017)

Blockchain and AI Security Opportunity

Blockchain and AI integration and innovation may present a more resilient and efficient path for decentralized cyber and physical devices to interactive, transforming modern infrastructure into array of smart autonomous systems of systems. Increased autonomy and control is essential to optimized the rapidly growing "Internet of Things" environment that Gartner has predicted to include 26 billion devices by 2020 (Gartner 2013).

"Simple and easy to write contracts appear to be sufficient for many entirely digital transactions. But as these systems start to interact with the physical world, there is likely to be a need for greater intelligence and real world knowledge in making decisions. AI systems will be needed to translate information from a wide variety of sensors into precise terms that smart contracts can act upon. In the other direction, contracts that lead to physical actions (such as delivery of items) will need to interface with human and robotic agents. For example, owner and operators of critical energy infrastructure might want insurance contracts against cyber-attacks and harmful weather conditions and a smart contract would need to determine when the payout event is triggered (Levine 2014).

Next Generation Energy Internet of Things Infrastructure

These grid optimization, automation and resilience improvements are essential operations and design criteria as we modernize our power grid. However, cybersecurity is often an afterthought as vendors and end users prioritize functionality and cost, leaving our power grid, the backbone of our economy, potentially vulnerable to a cyber-attack. This is especially true at the grid's edge which continues to increase the size and speed of data being collected and exchanged in absence of clear cybersecurity and IoT standards and regulation. Thus, the grid lacks the necessary defenses to prevent disruption and manipulation of DERs, grid edge devices and associated electricity infrastructure. Moreover, as the smart grid increases its connectivity and communications with buildings, cyber vulnerabilities will extend behind the meter into "smart" buildings, which also have a host of documented cybersecurity vulnerabilities.

Blockchain technology can also be applied to the smart grid to help reduce costs by cutting out 3rd parties and increasing the arbitrage opportunity for individuals to produce and sell energy to each other. Smart contracts facilitate peer-to-peer energy exchanges by enabling energy consumers and procures to sell to each other, instead of transacting through a multi-tiered system, in which distribution and transmission system operators, power producers, and suppliers transact on various levels (Mylrea and Gourisetti 2017). In April 2016, one of the first use cases was demonstrated where energy generated in a decentralized fashion was sold directly between neighbors in New York via a blockchain system, demonstrating that energy producers and energy consumers could execute energy supply contracts without involving a third-party intermediary; effectively increasing speed and reducing costs of the transaction (PWC 2017).

In addition to potential cost savings, transaction data might be more secure through decentralized storage and multifactor verification of transactions in the blockchain distributed ledger (PWC 2017). Blockchain reduces the need for 3rd parties to process transactions: Electricity is generated → Consumer buys the electricity → blockchain based meters update the blockchain, creating a unique timestamped block for verification in a distributed ledger: 1) At the distribution level, system operators can leverage the blockchain to receive energy transaction data to charge their network costs to consumers; 2) Reduces data requirements and increases speed of clearing transactions for transmission system operators as transactions could be executed and settled on the basis of actual consumption (Mylrea and Gourisetti 2017).

Smart contracts execute and record transaction in the blockchain load ledger through blockchain enabled advanced metering infrastructure (AMI). Blockchain based smart contracts can facilitate consumer level exchange of excess generation from DERs, EVs, etc. This could provide additional storage and help substation load balancing from bulk energy systems. Moreover, smart contract data is secured in part through decentralized storage of all transactions of energy flows and business activities. This highlights the disruptive potential for blockchain on energy markets through the introduction of a more autonomous and decentralized transaction model. This peer to peer system may reduce or even replace the need for a meter operator if the meter blockchain is shared with the distribution system operator.

Currently, the power grid lacks the necessary security and resilience to prevent cyber-attacks on DERs, grid edge devices and associated electricity infrastructure. Cyber vulnerabilities and interoperability challenges also extend behind the meter into building automation and controls systems. Applying blockchain could help increase fidelity and security of buildings to grid communications. Moreover, multiple customers can leverage the same widely witnessed blockchain to cryptographically verify the other entities data when needed, creating a distributed trust mechanism. Blockchain may also help solve several optimization and reliability challenges that have been ushered in with grid modernization.

Currently, time-lags for payment and uncollected bills leaves value on the table and the real cost associated with the energy value chain is not captured. Blockchain can record real time net loads and smart contracts execute customers distributed generated sales and purchases. Currently, grid operators lack visibility and control of real-time power flows and injections from DERs and distributed generation customers. Blockchain can help optimize network data and record residual energy at the substation level. Increasing the fidelity and control of utility data will also help settle with bulk systems as well as negotiate future contracts.

Conclusion and Future Research

Blockchain, AI and IoT have a lot of buzz right now. Reading the news one might assume that blockchain is a panacea for all that ills us – climate change, cyber security, volatile financial systems. AI articles suggest that robots are coming and may take our jobs. Internet of Things or IoT cyber incidents remind us that everything is increasingly connected to the internet and collecting and exchanging data that is potentially vulnerable. While these are disruptive in their own way and create some exciting new opportunities, many challenges remain. Several fundamental policy, regulatory and scientific challenges remain before blockchain realizes its disruptive potential. This sections explores some of the challenges as they relate to block chain's application to the array of things

Applying AI Blockchain to modernize electricity infrastructure also requires speed, agility and affordable technology. AI enhanced algorithms are not always cheap and often require prodigious data sets that must be broken down into a code that makes sense. However, there is a lot of noise or distracting data being exchanged in electricity infrastructure, making it difficult to identify what caused an anomaly – what is a software hire, cyber-attack, weather event, all the above? It can be very difficult to determine what normal looks like. Thus, developing an AI enhanced grid requires breaking down the data into observable patterns, which is also very challenging from a cyber perspective as threats are complex, non-linear and evolving.

New blockchain opportunities are also accompanied by the lack of policy, legal and regulatory frameworks. For example, even if some intermediaries are replaced in the energy sector, there still needs to be schedule and forecast submitted to the transmission system operator for electricity infrastructure to be reliable. Another challenge is incorporating individual blockchain consumers into a balancing group and having them comply with market reliability and requirements and submit accurate demand forecasts to the network operator. Managing a balancing group is not a trivial task and could potentially increase costs of managing the blockchain. To avoid costly disruptions, blockchain autonomous data exchanges, such as demand forecasts from the consumer to the network operator will need to be stress tested for security and reliability before deployed at scale.

Applying blockchain to modernizing and secure electricity infrastructure also presents several cyber security challenges. For example, Ethereum based smart contracts provide the ability for anyone to write electronic code that can be executed on a blockchain. For example, an energy producer or consumer agrees to buy or sell renewable energy from a neighbor for an agreed upon price that is captured in blockchain based smart contract. AI could help increase the efficiency and automate the auction to include other bidders and sellers in a more efficient and dynamic way, but this would require a lot more data and analysis of that data to recognize discernable patter in that data to inform the AI algorithm of the smart contract.

This also requires the code of the blockchain to be more resilient to cyber-attacks. Previously, Ethereum has shown to have several vulnerabilities that may underline the trustworthiness of this transaction mechanism. Vulnerabilities in the code have been exploited in at least three multi-million dollar cyber incidents. In June 2016, DAO, was hacked exploiting vulnerable smart contract code and extracting approximately $50 million dollars. In July 2017, vulnerable code in am Ethereum wallet was exploited to extract $30 million dollars of cryptocurrency. In January 2018, hackers stole roughly 58 billion yen ($532.6 million) from a Tokyo-based cryptocurrency exchange. Coincheck Inc. This incident highlighted the need for increased security and regulatory protection for cryptocurrencies and other blockchain applications. The Coincheck hack appears to have exploited vulnerabilities in a "hot wallet" which is a crypto currency wallet that is connected to the internet. In contrast, cold wallets, such as Trezor and Ledger Nano S, are cryptocurrency wallets that are stored offline.

Despite being a centralized currency, Coincheck was a centralized cryptocurrency exchange with a single point of failure. However, the blockchain shared ledger of the account may potentially be able to tag and follow the stolen coins and identify any account which receives them (Fadilpašić and Garlick 2018). Storing prodigious data sets that constantly growing on a blockchain can also create potential latency or bloat in the chain, requiring large amounts of ram and memory on a server. These requirements for ethereum based smart contracts have grown over time and the block takes a longer time to get processed. For time, sensitive energy transactions this may create speed, scale and cost issues of the smart contract is not designed properly.

References

Fadilpašić, S., Garlick, S. (2017, 1/26), "Coincheck Hack: "The Biggest Theft in the History of the World" Crytonews.

Franco, P. "Understanding Bitcoin: Cryptography, Engineering and Economics", John Wiley & Sons. p. 9, 2014.

Johnson, M. Guardtime, 2017 "Keyless Signature Infrastructure (KSI) Overview" Guardtime Publication

Lawless, W. 2017. "Artificial Intelligence, Blockchain and Redundancy" Email Exchange.

Levine, A. (2014). Application specific, autonomous, self-bootstrapping consensus platforms. Retrieved from https://bitsharestalk.org/index.php?topic=1854. 0

Mylrea, M, Gourisetti, S. 2017. “Leveraging AI and Machine Learning to Secure Smart Buildings”, Book Chapter in AAAI, Stanford University, Springer

Mylrea, M, Gourisetti, S. 2017. “Blockchain: A Path to Grid Modernization and Cyber Resiliency,” North American Power Symposium,

M. Mylrea, S. Gourisetti, R. Bishop, M. Johnson, "Keyless Signature Blockchain Infrastructure : Facilitating NERC CIP Compliance and Responding to Evolving Cyber Threats and Vulnerabilities to Energy Infrastructure", IEEE PES T&D Conference and Exposition, 2018 (under review)

Mylrea, M. 2017. Smart Energy-Internet-Of-Things Opportunities Require Smart Treatment of Legal, Privacy and Cybersecurity Challenges. Oxford Journal of World Energy Law & Business

PWC Global Power and Utilities, 2017. “Blockchain opportunity for energy producers and consumers.”

Tapscott, A. “The Blockchain Revolution: How the Technology Behind Bitcoin is Changing Money, Business, and the World”, Portfolio, 2016.

The Smart Data Layer

Magnus Sahlgren,[1⋆] Erik Ylipää,[1] Barry Brown,[2] Karey Helms,[3]
Airi Lampinen,[2] Donald McMillan,[2] Jussi Karlgren[3]
[1]RISE SICS, Kista, Sweden
[2]Stockholm university, DSV, Kista, Sweden
[3]KTH, Stockholm, Sweden
[⋆]Corresponding author: magnus.sahlgren@ri.se

Abstract

This paper introduces the notion of a *smart data layer* for the Internet of Everything. The smart data layer can be seen as an AI that learns a generic representation from heterogeneous data streams with the goal of understanding the state of the user. The smart data layer can be used both as materials for design processes and as the foundation for intelligent data processing.

IoT and Interaction

One of the more ominous visions of the future Internet of Everything (IoE) is a swarm of loosely integrated systems (e.g. the smart home, social media apps, health and fitness wearables, etc.) that constantly crave our attention with applications bombarding us with notifications and alerts, and devices demanding administration and care. Rather than improving quality of life and efficiency of work, such excessively attention-seeking technology will lead to cognitive overload, adding both stress and complexity to everyday life. The main problem, and risk factor for such a future technological dystopia, is that different forms of smart technology do not blend and cannot interface with one-another, and most importantly, end-users have to learn how to interact with each of the different systems, one by one. In some sense, this is like personal computing before the desktop metaphor, the Internet before the web, or mobile computing before touch interfaces. In short, Internet of Things (IoT) (and IoE) lacks an appropriate interface paradigm.

As one step towards a solution to this interface problem, we investigate the possibility of defining and applying a *smart data layer* that integrates heterogeneous data streams into a coherent representation that can serve as the foundation for further, intelligent, data processing. The idea is not to provide a uniform communication protocol between applications and devices, but to provide a representation of the *state of the user*, to enable more intelligent interface design. The problem we would like to mitigate is for applications and devices to know *when* and *how* to interact with the user. As a simple example, if the user is in a very agitated state, we probably should not send loud audible notifications that the milk in the refrigerator is almost finished and needs refilling, or for that matter send intense tactile vibrations indicating that the user has been stationary for too long and that it is now time to get up and move. Our vision is a data layer that *learns* from the user's behaviors, and that is empathetic to both the current state of the user and the current state of the system. This position statement describes our current research path, and provides some background and motivation for the smart data layer.

AI and Representation Learning

AI will be a critical component in the development of IoT and its various flavors, not only for making sense of the interconnected systems, but also – and equally important – for making sense of the user of the system. The ultimate goal is to *understand* the user; where is the user, what is the user doing, how is she feeling, what are her goals? In short, what is the *state* of the user? Note that we use the term "state" in a broad sense; it can encompass anything from a geographical location, to a task, to the emotional state of the user, to a prediction of the user's next action.

Solving individual tasks such as locating the user, classifying her behavior, or detecting her sentiment are interesting, and potentially useful, tasks in their own right, but they require an ontology to start from. We have to know which are the possible locations, behaviors and sentiments in order to determine which of them the user belongs to. Defining or acquiring such ontologies is typically a task-specific problem, as is the optimization of classifiers. We do not believe that we (at present) can design or learn one generic ontology and one generic classifier that can solve any problem. However, we do believe that we can learn one generic *representation* that can be common for all these problems. Ideally, this representation will capture the causal factors of variation in streaming data of different modalities and rates.

The idea of a generic representation that can be used for various different purposes is not novel in itself, see Bengio et al. (2013) for a review. A good representation simplifies tasks, and a desirable property of a representation is the separation of the causal factors that gives rise to a phenomena. Digital images are an example of representations that are difficult to use directly for solving computer vision tasks. The pixels in the two-dimensional grid explain very little of the scene that generated them. A representation that directly encodes the objects in the scene, their state and surroundings would make automatic decisions based on the scene much simpler. *Representation learning* can be thought of as a generalization of this inverted rendering process, where we infer what the causal factors were that generated the data using methods from statistical learning. Popular methods include latent variable models, deep neural networks and compressed sensing.

Several researchers have published papers in this research direction; one example is Collobert et al. (2011), who propose a unified neural network architecture that can be applied to natural language processing and learns shared representations of language useful for solving a variety of tasks. The field of deep learning to a large extent embodies the idea that it is possible to learn a compositional, generic representation that can be used to solve many different problems; in image recognition for example, it has become customary to use the unit activations of deep neural networks trained on very large datasets (such as AlexNet (Krizhevsky, Sutskever, and Hinton 2012) or ResNet (He et al. 2016)) as the basic representation when building novel classifiers. This method of *transfer learning* is useful where representations learned on large data sets can be used to solve related tasks where data is scarce, see Oquab et al. (2014).

Another recent example of representation learning is the StarSpace framework of Wu et al. (2017), which is a general-purpose neural network representation that can solve a wide variety of problems.

(Word) Embeddings as a Starting Point

Our vision of the smart data layer builds on the prior art discussed in the previous section, and is inspired by the development of *word embeddings* for natural language processing (Turney and Pantel 2010). Embeddings are low-dimensional representations that compress and encode co-occurrence information from the input data. A co-occurrence event is simply the simultaneous occurrence of two (or more) variables. In language data, the variables are typically words, and a co-occurrence is simply a sequence of words. The point of embedding co-occurrence information in a low-dimensional representation is that the resulting representation generalizes from the observed co-occurrence events, and enables quantification of *distributional* (as in a word's distribution over the data) similarity. Since distributional similarity is a proxy for semantic similarity, embedding models can be seen as computational models of meaning (Sahlgren 2006).

Word embeddings have become ubiquitous in both natural language processing and machine learning. However, current embedding models rely on a severely limited, and somewhat naïve, ontology. Most current models are confined exclusively to text data, with words being the only linguistic items under consideration. The fact that two words tend to co-occur is admittedly a useful clue to the meaning of the words, but there may be other types of contextual information that can provide equally useful clues for modeling meaning. In natural discourse, tone of voice, gestures, facial expressions, even time and location are important contextual factors that influence semantic processing. It seems reasonable to assume that this should apply also to computational models and AIs that aim to learn language.

Some recent studies have begun to investigate the possibility to extend the ontology of the co-occurrence model with other modalities such as vision and sound (Bruni, Tran, and Baroni 2014; Vijayakumar, Vedantam, and Parikh 2017). Our aim is more ambitious; one of the goals of the smart data layer is to extend current representation learning models with multi-modal contexts that encompass not only vision and sound, but also other types of contextual data, such as spatio-temporal information, various types of sensor data and infrequently occurring instantaneous events. If our ultimate goal is to build true AI, its representation must be built from more senses than just text.

The Data Sandbox

The type of representation learning mechanisms discussed in the previous sections are data-intensive and require large amounts of data to learn from. We expect the future IoTs to produce tsunamis of data where such models will thrive. However, getting access to such amounts of controlled data for development purposes is currently more difficult. We use the notion of a *data sandbox* (indicating that we start with baby steps) for collecting heterogeneous multimodal data. The data sandbox collects information from a user's computer, and stores the following information:

- Text on the user's screen (using the Google Cloud Vision API[1]).
- Text from the user's keyboard.
- Transcribed speech (using PocketSphinx[2]).
- Sentiment based on the user's facial expression (captured by the computer's camera, and using the Google Cloud Vision API).
- Sentiment based on faces on the user's screen (using the Google Cloud Vision API).
- Labels and categories recognized on the user's screen (using the Google Cloud Vision API).
- Various sensor data, including:
  - CPU usage.
  - Memory and disk usage.
  - Battery life.
  - Temperature.

This heterogeneous data will serve as the foundation for our initial experiments on multimodal representation learning. The idea is to extend embedding models with extralinguistic contexts, such as sentiment labels from facial expressions, or even CPU usage and core temperature. Although the amount of data that we expect to be able to collect using the data sandbox is too small to allow for more advanced techniques such as deep learning or compressed sensing, we plan to use statistical correlation measures to find interesting patterns in the data. As an example, imagine that we

[1] https://cloud.google.com/vision/

[2] https://github.com/cmusphinx/pocketsphinx

use a word embedding technique to learn a concept such as `soccer` based on the text on a person's screen.[3] Next, imagine that we notice that the `soccer` concept often co-occurs with positive facial expressions captured by the computer's camera and low CPU usage. This pattern constitutes a higher-order concept, which we might label something like `taking a break from work`. If instead we notice both high CPU and memory usage in conjunction with the `soccer` concept, we might instead infer that the user is in a state of `waiting for the experiment to finish`.

Possible Use Cases

Producing representations of diverse, textual and non-textual, data provides the possibility to represent user activity in diverse and interesting ways. Yet how could this be made actionable to have an influence on user or system behavior?

One approach taken by Intelligent User Interface research has been to make use of Bayesian models of user activity, "automatically" activating system actions based on predicted desired user outcomes. This has been used to, for example, allow systems to achieve a basic understanding of user intention based on context, and to perform different actions at different times in response to the same input (Wilson and Shafer 2003). Other work has made use of one of our data streams (text scraped from the user interface) to predict users' ongoing "tasks". While potentially interesting, this is a heavily reductionist model of user activity, and user state more broadly which is multifaceted. Clearly, a general representation of user activity has the potential to work in more complex ways.

In conceptualizing different uses of the representations, we have worked with open concepts applicable to varied contexts. Taking a historical view of context, distinctive ways of visualizing user activity from the data streams collected could support searching of past activity by users through looking for similarities between current activity and past activity events. The same interaction paradigm could afford the exploration of activity between users, or groups of users, and could be expanded to not only show the temporal relationship to the membership of a particular class of user, but with expanding the interface to expose the dimensions which relate to each classification. This could show that in one dimension an individual may be an outlier, but similar to many others in the rest of the vector representing this user.

Diverse representations might also enable a richer understanding of contextual inactivity and object appropriation. The insights into user relationships with and through things that this would provide could allow for the development of more sustainable products and systems. A deeper understanding of relationships might also inform a more meaningful design of interactional dialogs with conversational and embodied agents that appropriately act and enact with users and on their behalf. Additionally, a smart data layer might also support the design of experiential narratives that assist multiple user intentions with multimodal interactions for immersive or embodied user experiences. More broadly, building representations of users' ongoing activity may provide ways of supporting ongoing activities, such as speech recognition, Internet search, and advertising. However, we suspect that this would not be in the classic prompting of activity, but in different classes of activity that fit more with the ongoing modeling of action. The openness of these initial concepts enables future investigations into specific use cases across many contexts, from idiosyncratic routines to affective health to enterprise workflows, in which *when* and *how* to interact with the user requires a careful consideration of what can be understood from the systems' understanding the user.

In conclusion, while many of these envisioned use cases for a smart data layer might break with the expected utilitarian forms of use, we propose that in designing such a layer to support more playful, meaningful, and contextually appropriate applications it can be a driver for the development novel paradigms of interaction for the Internet of Things.

Acknowledgments

This work is funded by the Swedish Foundation for Strategic Research (RIT15-0046).

References

Bengio, Y.; Courville, A.; and Vincent, P. 2013. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* 35(8):1798–1828.

Bruni, E.; Tran, N. K.; and Baroni, M. 2014. Multimodal distributional semantics. *Journal of Artificial Intelligence Research* 49(1):1–47.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12:2493–2537.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In Pereira, F.; Burges, C. J. C.; Bottou, L.; and Weinberger, K. Q., eds., *NIPS 25*. Curran Associates, Inc. 1097–1105.

Oquab, M.; Bottou, L.; Laptev, I.; and Sivic, J. 2014. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1717–1724.

Sahlgren, M. 2006. *The Word-Space Model*. Ph.D. Dissertation, Stockholm University.

[3] Such concept learning could be accomplished e.g.by clustering the words in an embedding model, resulting in clusters of words that have a semantic relation. A soccer cluster might be populated by words such as "offside", "ball", "goal", "kick" and "Zlatan" (the name of a famous Swedish soccer player).

Turney, P. D., and Pantel, P. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research* 37(1):141–188.

Vijayakumar, A.; Vedantam, R.; and Parikh, D. 2017. Sound-word2vec: Learning word representations grounded in sounds. In *Empirical Methods in Natural Language Processing (EMNLP)*, 931–936. Association for Computational Linguistics.

Wilson, A., and Shafer, S. 2003. Xwand: Ui for intelligent spaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '03, 545–552. New York, NY, USA: ACM.

Wu, L.; Fisch, A.; Chopra, S.; Adams, K.; Bordes, A.; and Weston, J. 2017. Starspace: Embed all the things! *arXiv preprint arXiv:1709.03856*.

The Web of Smart Entities
Towards a Theory of the Next Generation of the Internet of Things

Michael Wollowski
Rose-Hulman Institute of Technology
5500 Wabash Ave.
Terre Haute, IN 47803, USA
wollowski@rose-hulman.edu

John McDonald, Vishal Kapashi, Ben Chodroff
Clear Object
8626 E 116th Street, Suite 300
Fishers, IN 46038
{john.mcdonald, vishal.kapashi, benjamin.chodroff}@clearobject.com

Abstract

We argue that the next generation of the IoT is about a web of smart entities. It is about real-time data. It is about data from various sources, from sensor data to data from smart entities. A key feature of the web of smart entities is accurate models which are continuously updated with live data. The models will be given authority to act and as such lead to yet unforeseen levels of automation. Models will be interacting with each other in more or less tightly coupled feedback loops, again, raising the level of automation. We see the smart entities as polite assistants, designed to make our lives more convenient. Something that will gracefully bow out, when asked to do so. In this context, we will address several modes in which to interact with and control the resulting automation.

Introduction

We believe that a large aspect of the next generation of the internet of things are learned models augmented with real time data that are authorized to act on our behalf. We explain and justify our vision in detail, as well as the resulting hyper automation.

(Gubbi et al. 2013) present a vision in which they emphasize the importance of cloud computing; we agree with their assessment. On page 1646, the authors state that ”This platform [i.e. cloud computing] acts as a receiver of data from ubiquitous sensors; as a computer to analyze and interpret the data; as well as providing the user with easy to understand web based visualization. The ubiquitous sensing and processing works in the background, *hidden* from the user.” Again, we could not agree more and explain in more detail what sort of processing may take place in the background.

(Weiser, Gold, and Brown 1999) defines a smart environment as ”the physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network.” We will generalize this to emphasize real-time data that enables one to build real-time models. In this context, we will argue that there is real-time data that comes from sources other than sensors.

(Stankovic 2017) sees a ”... significant qualitative change in how we work and live.” We will expose some of those changes and further refine his assessment. He continues by stating that ”We will truly have systems-of-systems that synergistically interact to form totally new and unpredictable services.” We agree with this assessment and attempt to shed some light on the kinds of services we may expect.

From a technical perspective, this paper takes off where the books ”The internet of things” (Greengard 2015) and ”Precision” (Chou 2016) left off. The IoT is an exponential technology. Now is the time to expose what will happen in the not so distant future. We need to debate and decide how our data gets used. Can we design devices and applications so that the entities which generate the data own the data and have complete, explicit control of it? Computer scientists and people interested in this technology need to work with businesses and governments to define standards and best practices for this vision to take off. For some aspects, we need to have a legal framework in place. From a perspective of analyzing the impact of the IoT, this paper continues to refine the ideas presented in the book ”How IoT is Made” (McDonald, Pietrocarlo, and Goldman 2015).

(Tucker 2014) and (Siegel 2016) touch on aspects related to the future of the IoT. They focus on big-data and predictive analysis. Predictive analysis can reveal things that may be shocking (Duhigg 2012). However, we focus on automation that results when models that learn specifics about someone or something’s behavior are empowered to act.

In this paper, we explain smart entities and present a comprehensive vision of the next generation of the IoT in the context of health. This enables us to analyze the sort of future we wish to formalize. It will be sufficiently powerful to generalize to other domains of the IoT. We present our theory of the next generation of the internet of things, which we see as a web of smart entities. We discuss the automation that results when models built on data are empowered to act and we discuss ways of acting in a hyper-automated world.

Smart things

It has been argued that the internet of things has a PR problem (Eberle 2016). Rather than talking about the IoT, we should be talking about smart things, such as smart cars or smart cities which are powered by the internet of things. We agree with this assessment and so do others (Bassi et al. 2013; Willems 2016). At the most basic, the IoT is about connecting all sorts of things to the internet. Those

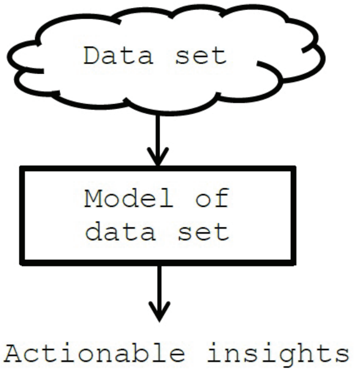


Figure 1: Current state of the art in the Internet of Things

things, whether washing machines, cars, our bodies or our food (Heikell 2016) produce data, in particular, real time data. Many times this data is useful on its own, for example a coffee machine might indicate that it is in the on position. This would be useful for anyone leaving for a vacation who is worried whether they left the coffee machine on.

Many items considered to be part of the IoT space can also receive input. Going back to the coffee machine, one could, or better yet, one's calendar could instruct the coffee machine to make coffee at some specified time. A more sophisticated example is advertised by Philips' Hue light bulb; it is designed to respond to mood information and the beat of the music one is listening to (Philips 2016). Since those devices can process information, they are oftentimes called *smart*, hence, smart homes and smart cities.

While many times, the data generated by devices is useful on its own, value and insight can be generated by building models of the data. At the most basic, a model of a sensor may be used to interpolate missing data or determine whether data is out of an expected range and as such may be faulty. At a higher level, models of data can be used to produce considerable value. Cummins Engines, the largest independent manufactures of diesel engines, uses telematics, i.e. real-time engine data to build real-time models of how their engines actually perform. These models are then used by Cummins in several ways. By running live engine data against the model, they can ascertain the general health of a particular engine. By using predictive analysis, Cummins is able to predict various scenarios ruinous to an engine and as such is able to alert fleet operators, in real time about fault-codes and their significance on the continued operation of the engine (CumminsEngines 2016).

We consider the Cummins example to be the state of the art with regard to current practice of the internet of things, in the sense that robust and repeatable solutions in this mold exist. This state of the art is captured in figure 1.

A Vision of the next generation of the IoT

The current generation of the IoT consists primarily of homogenous systems, i.e. systems that typically do not interact with systems external to them. For example, a home security system might interact with several sensors and perhaps with a dedicated sensor of another company's system. However, typically interactions with systems of other companies are limited and form an exception rather than a rule.

In the next generation of the IoT, we see many different systems interact to produce data and information. They will be used to seamlessly manage many aspects of businesses and of people's lives. They are in essence heterogeneous systems. Perhaps the best way to characterize the next generation is by describing a rich extended example. We pick the domain of personal health. While the next generation of the IoT will impact all aspects of people's lives, this domain is sufficiently complex to expose pertinent aspects of the web of smart entities. We should point out that the future of the IoT cannot be seen in isolation; it is imperative that advances in the IoT be seen in the larger context of advances in technology in general. This includes fields such as sensors, miniaturization of chip technology, cloud computing and Artificial Intelligence.

A Future scenario of IoT and health

In this section, we envision a future in which a person's health is maintained at an optimal level. We will address the following aspects of health maintenance: monitoring the body with Nano and Macro devices, diet, exercise, sleep, mental and physical health, health care as well as the use of artificial intelligence and cognitive assistants to enable physicians to make proper diagnoses and recommendations for treatment options. There will be many tight feedback loops.

Exercise The IoT made great strides measuring exercise activities. Additionally, many wearables can automatically sync exercise data to various web-sites. It is fair to state that a small set of wearables will enable a typical user to record an accurate picture of their exercise activities.

Diet When it comes to entering diet information, much of the data entry is manual at this time. Websites such as myfitenesspal.com take advantage of the fact that many people are creatures of habit. They simplify the data entry process by giving the user the ability to select from prior entries rather than having to re-enter detailed information about a dish. Another way to automate the process of maintaining diet information is by tying a meal planner to a site that maintains information about a person's diet. Websites like yummply.com offer diet information associated with a recipe. We imagine that restaurants, by way of an itemized bill augmented by nutrition information, will soon enable the automatic entering of diet information to diet management software. Think of augmenting Expensify.com with diet information and a plug-in for your myfitnesspal.com account.

Feedback loop: Fitness Given diet and exercise data, we can now determine whether a targeted balance of exercise and diet has been reached. Websites such as fitnesspal.com keep track of past exercise and diet activities and use various graphics to indicate the degree to which they are balanced. We now have a basic model of a person's physical fitness. In the future, we see models that are empowered to act, either by us, or by entities external to us, such as health insurance companies. We envision that by taking advantage of automated devices and the IoT, the model is asked to enforce dietary restrictions. For example, it could refuse to preapprove a meal in a restaurant that is judged as not fulfilling set dietary goals. Alternatively, the model could suggest a walk or bike ride instead of the use of a car or public transportation. It could go as far as asking the car to refuse to start. Alternatively, a doctor's visit might be scheduled, if fitness goals are not met in a consistent fashion. In the extreme, insurance rates may go up or the person may be loose health insurance coverage entirely.

Recommender system Given models of people's behavior, we are in a position to make recommendations. Right now, the yummly.com web-site makes recommendations based on preferences entered by the user. We imagine that in the future, recommendations can be made based on matching a user's recipe usage to those of others. This would be similar to how Netflix and Amazon.com recommend movies and goods. Similarly, based on a user's exercise patterns, and patterns that are similar to them, we imagine recommendations for modifications, additions or substitutions of exercise regimes.

Sleep Sleeping takes up about one-third of people's lives. We know that sleep deprivation is a known form of torture. As such, it is important to arrange that one gets sufficient sleep. With the creation of smart beds and wearables, it is possible to monitor people's sleeping patterns. A model of sleeping patterns informs whether one is getting enough sleep each night.

Feedback loop: Restful living The sleep model can interact with several systems in an attempt to regulate sleep. For example, it could be empowered to regulate the temperature in the bedroom. Additionally, it could interact with the meal planner to detect foods or drinks that are not conducive to sleep. In this context, it could merely inform the user or it could be empowered to remove or rescheduled such items to earlier in the day. The sleep model could interact with the calendar to perhaps move certain kinds of physical exercises that are detrimental to sleep.

Mental health We know that mental health is as important as physical health. The IoT will enable us to monitor and gauge our mental health as well. There are several aspects that can be measured: kinds and duration of mental activities. For example, by consulting a person's calendar, or some equivalent activities log, one can determine whether someone reads books, completes puzzles, engages in social activities, or has other creative pursuits. We envision that someone will soon develop a working laugh-o-meter app for smartphones, providing useful information on a person's mental health.

Feedback loop: Mens sana in corpore sano With an adequate model of people's mental health, one can now develop a more complete model of a person's health. Similar to fitness models, initially, a health model will likely just report on a user's health balance and may make recommendations. However, here too, we see a large potential for automation. This may be as simple as dynamically injecting physical or recreative mental exercises into a person's calendar, based on real-time data. In this context, we know that physical activity has a significant impact on mental health, as such; there is another feedback loop at play.

Physical health To add to exercise data, there are sensors such as pulse monitors, blood pressure monitors, wireless scales which give a fairly clear picture of many people's general health. If we include implanted devices, such as defibrillators, pace makers and blood glucose monitors, a good picture of physical health emerges even for people with major illnesses. Looking ahead, people proposed Nano-devices (Akyildiz, Jornet, and Pierobon 2011) which when placed in the body can provide more fine-grained monitoring of people's health or can be used to treat diseases such as cancer (Gaudin 2009). In this context, people are investigating challenges and opportunities of connecting Body Area Networks and other external gateways with in-body Nano-devices (Dressler and Fischer 2015). Nano-devices are expected to communicate, among others, on the molecular level. This communication has a very high latency time, something on the order of 12 hours and it is not necessarily reliable. As such, multiple Nano-devices would be used so as to get a more reliable picture. Think of how google maps aggregates data to get reliable information for traffic flow.

Automatic scheduling of doctor visits Combining a real-time accurate model of physical health with best practices in health care, we imagine that the model will be empowered to make appointments with various health care professionals as necessary. Conversely, some office visits will likely be eliminated completely. Many times, when our children are ill, we know that they need an antibiotic. Perhaps the systems and the regulations about prescribing medication will change so that some medication can be prescribed based on real-time data and best practices. An interesting side effect of both scenarios would be the effect it would have on how doctors and healthcare professionals spend their time. According to the New York Times, doctors find it hard to spend more than 8 minutes per patient visit (Chen 2013). With the ability to measure blood pressure, weight, run blood tests and other simple tests through connected devices and possibly even get prescriptions based on these tests, there will likely be a drop-off in patient visits. This will allow doctors to spend more time with patients who have serious illnesses.

Epidemics Automatic collection and consolidation of health data will enable public agencies to detect developing trends in real-time (Jalali, Olabode, and Bell 2012). A crucial benefit in formulating a response, as in those situations, time is of the essence.

Emergencies With real-time data, we can imagine the automation of emergency responses. This data may be sourced from wearable devices or from devices external to us. Consider a car crash, based on data from wearable as well as telematics of all involved parties, the severity of a crash can be assessed and the need for medical assistance evaluated.

If emergency assistance is deemed necessary, pertinent information about the patient should be sent to the paramedics and the person's physical health model should interact with the hospital's scheduling system. Finally, the model could alert family members and co-workers.

Reasoned Input Another form of real-time data is input by a health care provider. While the recommendation, i.e. data provided by a physician is not as frequent as that of a, say, a wearable device, it nevertheless is real-time data. This is particularly true with advancements of tele-medicine. In this case, the input to the physical health model comes from an informed actor. Another kind of input may come in the form of revised nutrition or exercise guidelines, such as issued by the U.S. Department of Health and Human Services.

Information and support In today's health care world, patients and physicians are seen as partners. Many patients want to know more about their condition or feel that they are in charge of their health. As such, we imagine that if a model determines that a person has a certain illness; it may make information about that condition available to that person.

Cognitive Assistants Cognitive assistants, as proposed by IBM (KellyIII 2015), are designed to digest vetted data to provide additional information. IBM sees cognitive assistants as ”wise counselors” (IBMWatson 2012) to experts, such as oncologists. As IBM sees it, ”IBM Watson, through its use of information retrieval and natural language processing, draws from an impressive corpus of information, including MSK [Memorial Sloan-Kettering] curated literature and rationales, as well as over 290 medical journals, over 200 textbooks, and 12 million pages of text. Watson for Oncology also supplies for consideration supporting evidence in the form of administration information, as well as warnings and toxicities for each drug.” (IBMWatson 2016). In addition to providing potentially better treatment options, information about treatment options informs the model of a person's health about potential side-effects and the likelihood for success. Both are important pieces of information as they may affect a person's physical and mental health.

Artificial Intelligence There are a few ways in which artificial intelligence techniques will be helpful. It is to be assumed that when gathering data from different scenarios to form an overarching model, that there will be inconsistencies. Detecting and possibly resolving such inconsistencies can be accomplished with AI techniques such as proof checkers. In the example about sleep, perhaps one person needs to have lower temperature than their partner's preferences. Such constraints could be resolved or at least smoothened through constraint satisfaction techniques.

Feedback loop: Overall health We have shown how multiple sources of real-time data are used to build real-time models, which are used to monitor various aspects of a person's health and manage some of those aspects as well as their overall health in a seamless fashion.

Feedback loop: Insurance companies We have alluded to bringing insurance companies into the loop. While people may not mind that those who exercise regularly obtain better insurance rates, the big question is how much information to share with health insurance companies, for fear that coverage may be dropped or that insurance rates increase in an unreasonable fashion. Assuming guaranteed health coverage, or even assuming that people by and large wish to lead healthy lives, a sophisticated model of a person's health with multiple real-time data based feedback loops will by and large ensure a healthy life. Assuming everyone is doing their part in staying healthy; a compassionate person might argue that their health care should be taken care of. A compassionate person, feeling thankful that they are healthy, might furthermore argue that people with serious pre-existing or inherited conditions should receive all the health care they wish to receive.

Towards a theory of the web of smart entities

In this chapter, we develop a theory of the *web of smart entities (WSE)*. We analyze the examples described in section 2 to justify the components of the theory. We show that this web is about real-time data, real-time models, interactions between them and models that are authorized to act. Additionally, we contrast the WSE and Big-data predictive analytics.

Real-time Data

Sensor data Without a doubt, a key aspect of the IoT and by extension, the WSE is real-time data obtained from sensors.

Aggregated Data If we look at Nano devices, due to their brittleness, one needs to rely on aggregate data submitted by them. Similar to how Google Maps ascertains traffic data, it is simply the aggregate of data from many sensors. This data is closer in precision to sensor data and it is automatically collected in continuous time.

Generated Data If we look at how diet data is inputted to systems; it is currently not generated by sensors. While some diet data is entered manually, it is possible to transfer data from existing resources, such as personal and external databases. If we look at diet data, even though it is not generated by sensors, it is still real-time data. It is just that most people do not eat continuously, rather at certain times of the day. If a meal planner is used, then some of the data is known ahead of time. Some of the data will be imprecise, due to the fact that portion sizes served at home are not standardized. Perhaps after a few manual inputs, the system learns about individual appetites. Restaurants, by and large, have standardized portion sizes.

Knowledge There are many sources of knowledge, from dictionaries to data inferred by various AI techniques. This data is perhaps as much removed from sensors as any data can be. This kind of data, especially data made available through dictionaries is updated in real-time also, think about Wikipedia. However, it is unlikely that this data gets pulled on a continuous basis. Nevertheless, it will be pulled when a need arises, such as when a person is diagnosed with a certain illness and wishes to know more about it.

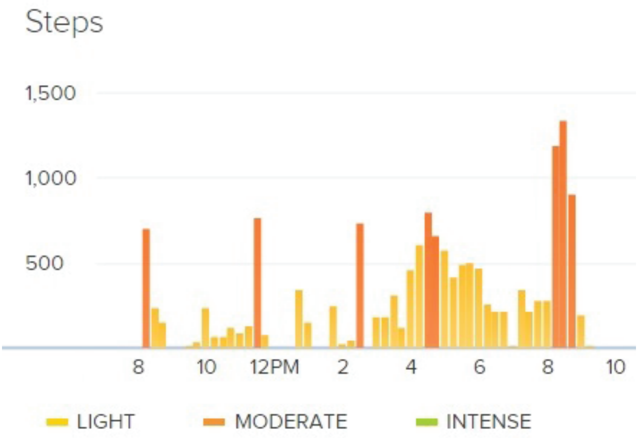


Figure 2: Exercise data

Human Input Human input may be as mundane as entering information that currently cannot be read by a sensor, to acknowledging information, to offering creative insight. If we look at a physician, many times, their diagnosis will be based on creative insight. While this data is not continuously generated as a sensor might, it is still real-time data. Similar to above, this kind of data is provided when the need for it arises.

Real-time, accurate models

The defining criteria of the web of smart entities are accurate, real-time models. In the past, models were built based on educated guesses, experience or historical data. With real-time data, models are continuously validated, refreshed and refined by live data as described in the prior sub-section. Equally important, the model building is automated.

Models Consider the data displayed in the histogram of figure 2. It is real-time data for number of steps taken over the course of a day. If we average this data over several days, assuming that the user has a fairly regimented exercise program, we will get a model of the exercise activities. Under the given assumptions, the model will be very similar to the histogram. If we assume that exercise patterns vary substantially, then a model that captures exercise activities by time of day may not be useful. Instead, one may have to be look for more coarse grained patterns or simply look at the total of exercise activities by day.

For a system that is balancing physical fitness, data on exercise and diet totals by day would be sufficient. For a system that is intended to be used to schedule exercise activities, a pattern of past behavior by minute or hour would be useful, if the person likes to have a regimented exercise pattern. If the person does not care, and this would be obvious from the data, then exercise activities could be scheduled at will.

Real-time models A real-time model is a model that is continuously updated by real-time data. We imagine that in some cases an overall average is desired and in other cases, a bias towards more recent data is desirable. Equally important to capturing real-time data are the definition of triggers that detect events. Such triggers can be defined in cases the real-time data deviates in noticeable ways from the data captured in the model. While some automation in the context of the web of smart entities will be driven by procedures that ingest data and models and act on the combined data, some automation will be caused by triggers which would then invoke dedicated procedures.

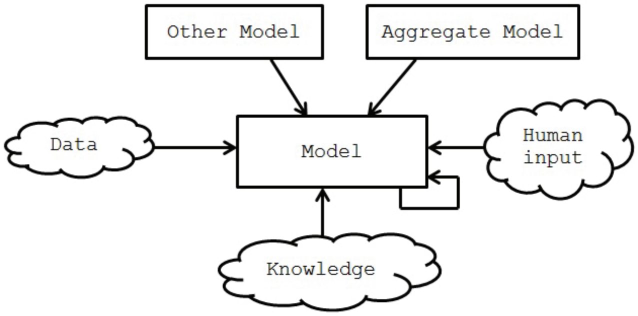


Figure 3: A model and its potential inputs

Complexity of models Since real-time models are data driven, models are as complex as the patterns in episodic data. We imagine that some patterns in the data repeat over and over again, while some might be considered an exception. Perhaps the best way to capture data patterns and exceptions to them is similar to how case-based reasoning (Wikipedia 2016) stores information. A collection of cases would be the model then.

Models produce data In this context, we see three additional data sources: other models, aggregate models and a feedback loop from the model to the model itself. Figure 3 shows the potential inputs to a model.

Other Models We have seen several examples of data that originates from models, such as diet and exercise data. A model that is interested in balancing diet and exercise would need to have access to those models. We imagine that the model which balances diet and exercise would furthermore interact with other models, such as calendars, cars, public transportation and restaurants. We should note that in this context, we use the term ”model” as shorthand for applications that maintain an underlying model of the data available to them.

Aggregate Models Just as Google aggregates data from individual cars, to construct a model of traffic congestion, we can imagine cases in which we wish to aggregate entire models. Consider models of exercise data. If we were interested in simply ascertaining the overall exercise activities of the employees, we would only need to gather a single data point of each employee. However, if we wish to ascertain exercise patterns, perhaps in the context of scheduling gym hours or to determine how big of a gym to build, then models of exercise patterns are necessary.

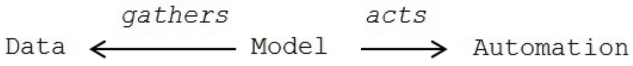


Figure 4: When models act, automation results

Feedback loop A feedback loop of a model to itself provides for the ability to fine-tune matching parameters. Suppose that a model of a person's food preferences is matched to someone else or to a small set of those of others. A recipe may be returned that the person likes. However, in case they do not like it, it may be useful for the model to provide additional matching parameters that may assist the recommendation system to determine better matches. We think of how case-based reasoning (Wikipedia 2016) matches new cases to the case base. We imagine that something similar might happen with prescription data. Based on actual use and the effects they have on a person, medications can be fine-tuned to a person.

Automation

In this section, we will explore the automation that results when real-time models are empowered to act. This scenario is depicted in figure 4.

Data validation The most basic form of automation is to remove data that is judged outside of an expected range and to complete data that is missing.

Managing learned behavior Once a model learned patterns of behavior, it can be authorized to act on someone's behalf. Suppose a model learned that every Tuesday evening, it is pizza night. Suppose it also learned that a given family always orders the same pizza. In that case the model can order the same pizza, to arrive at the usual time. To look at a more complex case, suppose that the model also learned that the given family never orders pizza twice in a row and that this family had pizza the night before. In that case, the model could ask for input, or perhaps act on some other learned behavior. Notice that in this case, the model acts on learned behavior, as well as real-time data.

Balancing A more complex use of a model comes about when models interact. For example, if the model of a person's exercise activities interacts with a model of a person's dietary intake, physical fitness can be balanced to specifications. If we empower the fitness model to make decisions, we can dynamically adjust a person's fitness. For example, the fitness model may encourage a walk or bike ride rather than the use of a car or public transportation. Perhaps, they alternatively recommend a dish that lowers a person's caloric intake.

Seamlessness There is data everywhere. In particular, it is likely that models will gather data about particular activities in different contexts. For example, food preferences should be gathered not just from meals prepared at home, but also from meals ordered at restaurants or consumed in other settings. This way, an overarching and more informed model can be built. Seamlessness comes about when an overarching model is applied in different contexts. If the model learned that someone likes their coffee black, then this is how it should be prepared, whether at home, at work, or by a coffee shop.

Recommendations Models of a person's behavior can be used to make recommendations based on matching to like models. For example, diet preferences, just like the preferences that Netflix and Amazon gather about their customers, can then be used to match similar models and based on those matches, recommendations may be made.

Out-of-the-box recommendations Not everyone is enamored by recommendations made by Netflix, and sometimes, one may be outright puzzled by them. In particular, the recommender systems described above are only as strong as the imagination of those who have models judged similar to the input model. An alternate way of making recommendations is by using AI techniques, such as IBM's proposed cognitive assistants. Assuming a model about some aspect of a person's life, AI techniques will be able to search for information that is not based on other people's behaviors.

Smart substitutions The use of AI technologies and the use of ontologies such as used in the context of the semantic web enable smart substitutions. We see examples of this when, based on dietary restrictions, alternate meals may be suggested, or when certain kinds of exercises are recommended based on availability or opportunity.

Interacting with automation

We described a highly automated world which is built on real-time data and models that are continuously updated and refined. It might be daunting to know that various computing environments record every activity and build various models about them; kind of an alter ego. Perhaps, the computing environment knows you better than you know yourself. This may be hard to take for many people. Will people feel watched? Will they feel "verklemmt?" How will all of this affect creativity? Will people hide things from the model or purposefully engage in activities to deceive it, as described in (Orwell 1950)? Will people get used to "big brother" watching them? To which degree does the automation limit what we can do, a point made by (Agamben 2010).

We attempted to give a reasonable view of the future, which we see as largely positive. We see the WSE as inhabited by polite assistants, designed to make our lives more convenient. We envision automated assistants that gracefully bow out, when asked to do so. As such, we envision, perhaps too hopefully, a future in which people can choose the level of interaction with the WSE. In particular, we would argue that the ability to choose the degree of automation should be a design feature, something that the user can explicitly manage and to a certain degree, something that the model anticipates. In the same context, the user should be able to control what information is gathered about them and who has access to it. Some models may not be as precise as they can be, however, they will be designed to perform tasks within the limits set by the user.

We describe three primary ways of interacting with automation: autonomous, semi-autonomous and manual. Undoubtedly, there will be a spectrum of interaction patterns.

Fully autonomous

In this mode of interacting with automation, the system makes all the decisions. For example, some people eat the same dish on specific days of the week. This is behavior that can be quickly learned. The meal planner can be authorized to order dishes or the ingredients for them and arrange for delivery at desired times (another learned behavior.) Similarly, some people always order the same dish at a particular restaurant. This behavior too can be quickly learned and applied appropriately. There are many other components of our lives that have little to no variation. Many people order the same toiletries, clothes, cars, take the same route to drive to work, have the same weekly work schedule, and engage in the same sort of recreational activities on a weekly basis. It is not unreasonable to assume that large swatches of our lives can be automated. The benefit of this mode is that it would take care of routine or nuisance activities.

On a side note, we recall a time when people first attempted to ”live off” the internet for a given period of time. In the same vein, it might be asked whether people would be able to live in fully autonomous mode. Many people are creatures of habit and as such there is no reason to believe that it cannot be done. Whether this would be an interesting life is another question.

Semi-autonomous

In this mode, the user gives some input to the model. In some cases, information will be requested, in other's the user will simply override certain inputs or parameters. The override may be as innocuous as not following the directions of a navigation system. When operating in this mode, we imagine that the input range will be limited to acceptable operating parameters. Think of how pilots of an Airbus, no matter what input they give, cannot put the airplane in a stall situation.

Another example of an interaction pattern in semi-autonomous mode is as follows. Suppose a cook heard about substituting riced cauliflower for rice in stir-fry dishes. The recipe may have to be adjusted to account for the riced cauliflower; however, all other aspects of the recipe are unmodified. If there is a recipe in some accessible data base that already accounts for the new ingredient, then it can be consulted. However, if there is none, then the automated pantry would be able to purchase all other ingredients in addition to the cauliflower. If the system is sufficiently knowledgeable, it may inform the cook that they may have to obtain an appropriate device to turn cauliflower into riced cauliflower.

Manual

In this mode, the user acts without the assistance of automation. There will likely be different flavors of this mode. One may enable the system to make suggestions. We are thinking of a collision avoidance system that gives an audible warning when it determines that a collision is imminent. Alternatively, one may ask the system to enforce certain boundary conditions. Going back to the example of a collision avoidance system, it would apply the brakes rather than issue a warning. This latter case is at the boundary to semi-autonomous interaction. We list it here, because the primary mode of interaction is manual. Finally, a user may simply tell the model to bugger off. This latter case may happen when driving in an emergency situation.

It is likely, that even in this mode, the model will gather data about the user. However, thought should be given to enabling the user to easily control which information is gathered about them and who can see it.

One may wonder about the feasibility of operating in ”manual mode.” In particular, it could be argued that it is necessary for people to act in ”manual” mode ever so often so as to maintain their ”edge.” Perhaps this is not as important for situations such as shopping; however, it might be important when skills are involved. For example, modern aircraft are highly automated to the extent that some pilots may not get sufficient practice for crucial maneuvers. There is talk about having pilots perform certain critical maneuvers on a regular basis so as to maintain their skills. In the same vein, perhaps people should be forced to ever so often live without the automation of models involving crucial skills.

Limits of Instrumentation

We know that there are limits to what can be instrumented, based on cost, practicality and user acceptance. Consider the task of monitoring sun spots on the skin for the purpose of detecting melanoma. It may be impossible or not acceptable for people to wear sensors. Perhaps a more acceptable solution exists in a smart mirror that can take images of a person's skin and send them to a server to be analyzed for melanoma. In addition, we know that big-data and predictive analysis can go a long way to infer data (Duhigg 2012).

Limits of Automation

Even in cases where instrumentation is possible and acceptable, there are instances where automation may not be desirable or feasible. Many people have a morning wake-up routine that while it can be automated, helps people to wake-up and start into their days. It is simply part of their psyche to follow their routine, however archaic it may seem.

To give an example of a case where automation is not feasible, consider the evening routine of young children. There is a clear routine to it and as such a model can easily be learned. However, at this point, there is only limited opportunity for automation. While lights may be adjusted, the bathwater can be drawn and reward points can be tallied, at the end of the day, it takes a tenacious parent to enforce rules the limits of which toddlers are so eager to explore.

Conclusions

In this paper we defined the next generation of the internet of things as a web of smart entities. We argued that in this web is about real-time data which originates from many sources, only some of them are sensors. We argued that a defining characteristic of the WSE are accurate real-time models which capture and describe the data. We argued that when models are empowered to act, an unprecedented level of automation will result. We depicted a world in which this automation will manage and arrange many activities which are generally considered nuisance activities. The effect is that people will be able to focus on things that are important to them. In a sense, it enables people to focus more on their lives than before. We portrayed three principle ways of interacting with models: fully autonomous, semi-autonomous and manual. We ended the paper with a brief discussion of the limits of instrumentation and automation.

Acknowledgements

This work was completed while the first author was on sabbatical at Clear Object.

References

Agamben, G. 2010. *On what we can not do, in: Nudities*. Stanford University Press.

Akyildiz, I. F.; Jornet, J. M.; and Pierobon, M. 2011. Nanonetworks: a new frontier in communications. *Communications of the ACM* 54:8489.

Bassi, A.; Bauer, M.; Fiedler, M.; Kramp, T.; van Kranenburg, R.; Lange, S.; and Meissner, S. 2013. *Enabling Things to Talk - Designing IoT solutions with the IoT Architectural Reference Model*. Springer Verlag.

Chen, P. 2013. *For New Doctors, 8 Minutes per Patient*. http://well.blogs.nytimes.com/2013/05/30/for-new-doctors-8-minutes-per-patient/?_r=0 (accessed 27.10.16).

Chou, T. 2016. *Precision: Principles, Practices and Solutions for the Internet of Things*. self-published.

CumminsEngines. 2016. *Connected Diagnostics The lifeline for your engine*. https://cumminsengines.com/connected-diagnostics (accessed 27.10.16).

Dressler, F., and Fischer, S. 2015. Connecting in-body nano communication with body area networks: Challenges and opportunities of the internet of nano things. *Nano Communication Networks Journal* 6(2).

Duhigg, C. 2012. *How Companies Learn Your Secrets*. http://www.nytimes.com/2012/02/19/magazine/shopping-habits.html?pagewanted=6&_r=2&hp (accessed 27.10.16).

Eberle, R. 2016. *The Internet of Things has a Vision Problem*. http://www.cio.com/article/3028054/internet-of-things/the-internet-of-things-has-a-vision-problem.html (accessed 27.10.16).

Gaudin, S. 2009. *Nanotech could make humans immortal by 2040*. http://www.computerworld.com/article/2528330/app-development/nanotech-could-make-humans-immortal-by-2040–futurist-says.html (accessed 27.10.16).

Greengard, S. 2015. *The Internet of Things*. The MIT Press.

Gubbi, J.; Buyya, R.; Marusic, S.; and Palaniswami, M. 2013. Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems* 29:1645–1660.

Heikell, L. 2016. *Connected cows help farms keep up with the herdt*. https://news.microsoft.com/features/connected-cows-help-farms-keep-up-with-the-herd/#sm.001npdttm13z6dn2spb2ce2sm2jay (accessed 27.10.16).

IBMWatson. 2012. *Assisting oncologists with evidence-based diagnosis and treatment*. https://www.ibm.com/developerworks/community/blogs/efc1d8f5-72e5-4c4f-99df-e74fccea10ca/resource/Case%20Studies/IBMWatsonCaseStudy-MemorialSloan-KettingCancerCenter.pdf?lang=en (accessed 27.10.16).

IBMWatson. 2016. *IBM Watson platform helps fight cancer with evidence-based diagnosis and treatment suggestions*. http://www.ibm.com/watson/watson-oncology.html (accessed 27.10.16).

Jalali, A.; Olabode, O. A.; and Bell, C. M. 2012. Leveraging cloud computing to address public health disparities: An analysis of the sphps. *Online Journal of Public Health Informatics* 4(3).

KellyIII, J. 2015. *Computing, cognition and the future of knowing*. http://www.research.ibm.com/software/IBMResearch/multimedia/Computing_Cognition_WhitePaper.pdf (accessed 27.10.16).

McDonald, J.; Pietrocarlo, J.; and Goldman, J. 2015. *How IoT is Made*. self-published.

Orwell, G. 1950. *1984*. Signet Classics.

Philips. 2016. *Your personal wireless lighting system*. http://www2.meethue.com/en-us/about-hue/what-hue-does/ (accessed 27.10.16).

Siegel, E. 2016. *Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die, 2nd Ed.* Wiley.

Stankovic, J. 2017. Research directions for the internet of things. *IEEE Internet of Things Journal* 1(1):3–9.

Tucker, P. 2014. *The naked future What happens in a world that anticipates your every move*. Current Publishers.

Weiser, M.; Gold, R.; and Brown, J. 1999. The origins of ubiquitous computing research at parc in the late 1980s. *IBM Systems Journal* 38(4).

Wikipedia. 2016. *Case-based Reasoning*. https://en.wikipedia.org/wiki/Case-based_reasoning (accessed 27.10.16).

Willems, C. 2016. *Cruising to Safer, Smarter Street*. https://blogs.cisco.com/government/cruising-to-safer-smarter-streets (accessed 27.10.16).

Beyond Machine Intelligence: Understanding Cognitive Bias and Humanity for Well-Being AI

Early Dementia Detection through Conversations to Virtual Personal Assistant

Saleh Ahmed, Mahboob Qaosar, Rizka Wakhidatus Sholikah,
Yasuhiko Morimoto
Graduate School of Engineering, Hiroshima University
1-7-1 Kagamiyama, Higashi-Hiroshima, Japan
Email: d16269, d172517, k170039, morimo@hiroshima-u.ac.jp

Abstract

Early detection of dementia are important because it can slow down the progress of the disease. One of the popular way to detect dementia is based on cognitive tests. The tests are usually done in the clinical setup with the help of a psychometrically trained examiner. Revised Hasegawas Dementia Scale (HDS-R) is one of the prominent screening tests for dementia. We propose a method for early dementia detection by using a Virtual Personal Assistant (VPA) on a computer that has a natural language user interface, such as Amazon Echo, Apple Siri, Google Home, Microsoft Cortana, Softbank Pepper, Sharp RoBoHon, etc. In our proposal, we consider HDS-R as a guideline to examine dementia. A VPA extracts the necessary features from the verbal and interactive response of the patient to compute the level of dementia. Such implicit checking is physically and mentally much comfortable for old people. We believe the proposed method will be able to contribute future society.

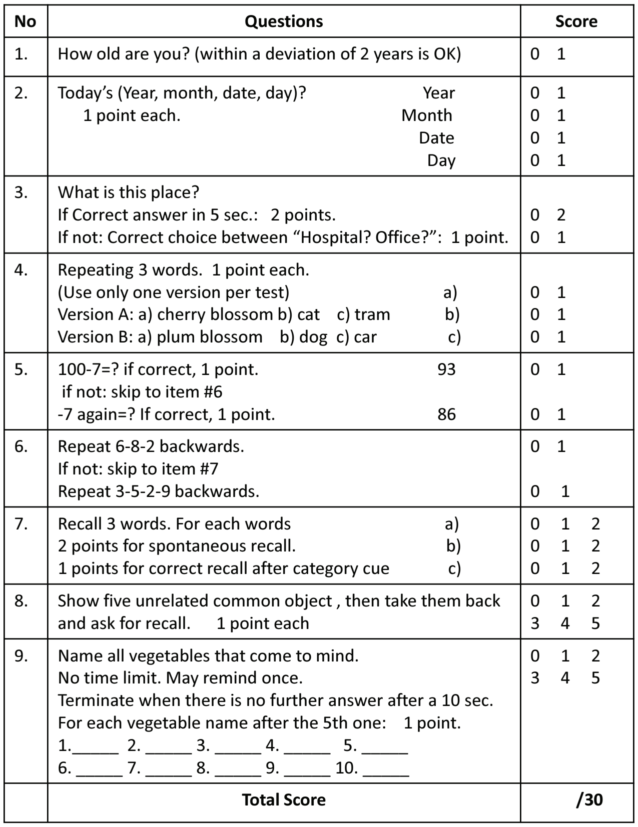
Introduction

Dementia is a category of brain diseases that cause a long-term and often gradually decrease the ability to think and remember. Dementia is classified as a neurocognitive disorder. It affects a person's daily functioning as well as other symptoms like an emotional problem, problems with language, and a decrease in motivation. More than 40 million people worldwide suffer from Alzheimer's disease, which is the most common dementia, and the number is expected to increase drastically in the coming years. But no real progress has been made in the fight against the disease since its classification more than 100 years ago.

Nowadays people are frequently using Virtual Personal Assistant (VPA) that has a natural language user interface, such as Amazon Echo, Apple Siri, Google Home, Microsoft Cortana, Softbank Pepper, Sharp RoBoHon, etc. Assistants can interact with people using voice and text to help them for finding web content, managing their daily routine, finding a route, purchasing right goods, help them in social conversation etc. Since VPA can interact with people, it can do the cognitive test to detect dementia. Revised Hasegawa's Dementia Scale (HDS-R) is a popular brief instrument for assessing dementia. We can conduct an automatic HDS-R test or collect the similar type of response from a spontaneous conversation with VPA.

Motivating Example

Y. Imai and K. Hasegawa (Imai and Hasegawa 1994) proposed revised Hasegawa Dementia Scale (HDS-R) and examined its usefulness as a screening test for dementia. In their study, they have suggested asking simple nine questions as in Figure 1 with different weight to a patient for investigating the possibility of having dementia. A description of each question is as follows.



No	Questions	Score
1.	How old are you? (within a deviation of 2 years is OK)	0 1

2.	Today's (Year, month, date, day)? Year
	1 point each. Month
	Date
	Day
		0 1
		0 1
		0 1
		0 1

3.	What is this place? If Correct answer in 5 sec.: 2 points. If not: Correct choice between "Hospital? Office?": 1 point.	 0 2 0 1

4.	Repeating 3 words. 1 point each.
	(Use only one version per test) a)
	Version A: a) cherry blossom b) cat c) tram b)
	Version B: a) plum blossom b) dog c) car c)
		
		0 1
		0 1
		0 1

5.	100-7=? if correct, 1 point. 93 if not: skip to item #6 -7 again=? If correct, 1 point. 86	0 1 0 1
6.	Repeat 6-8-2 backwards. If not: skip to item #7 Repeat 3-5-2-9 backwards.	0 1 0 1

7.	Recall 3 words. For each words a)
	2 points for spontaneous recall. b)
	1 points for correct recall after category cue c)
		0 1 2
		0 1 2
		0 1 2

8.	Show five unrelated common object , then take them back and ask for recall. 1 point each	0 1 2 3 4 5
9.	Name all vegetables that come to mind. No time limit. May remind once. Terminate when there is no further answer after a 10 sec. For each vegetable name after the 5th one: 1 point. 1._____ 2. _____ 3. _____ 4. _____ 5. _____ 6. _____ 7. _____ 8. _____ 9. _____ 10. _____	0 1 2 3 4 5
	Total Score	**/30**

Figure 1: HDS-R evaluations process

Question 1 [Age]: In this question, the examiner asks about the examinee's age if he can able to answer correctly within a deviation of two years get one point.

Question 2 [Orienta In time]: Here, an examiner actually asks about current year, month, day and name of the day of the week. The examinee gets one point giving each right answer.

Question 3 [Orientation in place]: In this section, the examiner asks the examinee about the place where actually he is now. If the examinee spontaneously gives correct answer get two points. His answer considers being correct if he able to understand where he is. If he fails to answer within five seconds gives him two options (hospital/house), at this point, give him one point for the correct answer.

Question 4 [Repeating 3 words]: Utter slowly one by one three words. A while later, ask the examinee to repeat the words. Score one to correctly utter each word. Teaching at least three times to memorize the words, if the subject not able to repeat. Remove the word from Question 7 as delayed recall word.

Question 5 [Serial subtractions of 7s]: This question is about the serial subtraction of 7 from 100. Firstly, ask the examinee to subtract 7 from 100. If his answer is right, give him one point and proceed to the second question. If the answer is wrong, then ask him next question 6. Ask him to subtract 7 from 93 as a second question. if his answer is right and gives him one point.

Question 6 [Digits backward]: For this question utter 3 digits, 6, 8 and 2, give one-second interval to each digit. At this point, ask the examinee to utter the digits backward. If the examinee able to repeat the first digit correctly, then go for the second digit and so on, give one point for each correct digit. If the examinee fails to answer this section proceed to question no 7. Otherwise, utter 3,5,2, and 9 in the same manner above. if the examinee can repeat backward correctly, give one point.

Question 7 [Recalling of 3 words]: The question no four is about repeating three words. At this point ask the examinee to recall the three words used in the question no four. Give two points to answer if it is spontaneous. If the examinee not able to answer properly, if the answer is cherry blossom give hins "It is a plant", if the answer is cat give hints "it is an animal", if the answer is tram give hints "It is a vehicle". Give one hint at a time and confirm examinee response.

Question 8 [Recalling 5 objects]: Here examiner uses five unrelated common objects. It may be a combination of a ring, a pen, a coin, a glass and a cigarette. Show the five object one by one, as well as uttering their name, after a while takes them back, give one point to each correct recall without considering the orders.

Question 9 [Generating vegetables]: This question is for observing generating fluency of an examinee. Ask the examinee for uttering the name of the vegetables, if the delay between any subsequent uttering of vegetable name is more than 10 second discontinue the question. After successfully uttering five vegetable name give one point for each vegetable up to tenth one.

These questions basically examine memory recalling and reasoning capability of the patient. They showed that HDS-R is able to screen dementia at the highest sensitivity of 0.90 and specificity of 0.82 at a cut off point of 20/21 of total score 30. The HDS-R can screen dementia at conceivable accuracy and efficiency and may serve to assess the severity of dementia changing with time and the effectiveness of pharmacotherapy and rehabilitation. The HDS-R will deserve intercultural application by virtue of universality of its contents. It will gain general acceptance from physicians because of its very simplicity with the utmost rationality and contribute to the everyday psychogeriatric management of demented patients. Also, it is always important to have additional diagnostic tools in arriving at an appropriate differential diagnosis for the memory impairment elderly

So, we can conduct HDS-R test or can find the similar type of responses in spontaneous conversation with VPA. For example, assume that a person asks "find me good hotels in bay area" to VPA. In response, VPA gives three names and ask to recall the three names. Such HDS-R related interactions are recorded to check dementia level of the person.

Related Work

Several works have been done and proposed for detection of dementia. Y. Imai and K. Hasegawa (Imai and Hasegawa 1994) revised their previous dementia scale and make several experiment to determine its usefulness as a screening test for dementia. J. B. Jimison et al. (Jimison et al. 2004) In this work, they monitor the interaction of the user with the computer and use this in the algorithm to evaluate cognitive performance. For this, they use a popular computer game usually played and enjoyed by elders who may have a risk for dementia. So, monitoring cognitive performance there are significant strategic planning in each level in the game. From this, they collect cognitive performance of everyone at frequent intervals. They monitor the movement of the mouse and their adaptation to the game difficulty to detect individuals cognitive performance.

In the research of E. M. Alkabawi et al. (Alkabawi, Hilal, and Basir 2017) proposed early detection of the different type of dementia, here they use the deep learning-based method in computer-aided diagnosis. They compare other three conventional computer-aided methods with their proposed method. They show that their proposed method produces a fair amount of accuracy to early detection of the different type of dementia. Y. Abe et al. (Abe, Toya, and Inoue 2013) illustrated how behavior sensing can be used for early detection of dementia. Here they use two different analytic methods to obtain data and use the data to find out the initial symptom of dementia. They use threshold based analytical method as the first one and use trend based analytical method as the second one. They use different scenarios and evaluate validity and immediacy of their proposed two methods. T. Shigemori et al. (Shigemori et al. 2016) proposed a new system that can quantitatively measure dementia with high accuracy. In their system, they measure the type and progression of dementia without awareness of the patient. So, they introduce a system that uses daily conversation, drawing, facial expression etc. to detect dementia. Here, the mainly rely on Clock Drawing Test (CDT). They used computer-based CDT for their proposed system. In their proposed method, they extract feature using Weighted Directed Index Histogram from given image then use support vector machine for classification. They showed that their proposed method on an average over ninety percent of dementia cases correctly. H. Tanaka et al. (Tanaka et al. 2017) proposed a new method to measure dementia automatically. They use a computer avatar that can interact with people. The avatar can use spoken dialog functionality and use this dialog to conduct spoken quarries as in mini-mental state examination such as the Wechsler memory scale-revised, and other related neuropsychological tests. They recorded spoken dialog of 29 participants, 14 of them have dementia and 15 of them are healthy. They extract the various audio-visual feature from these individuals. Two machine learning algorithms (support vector machines and logistic regression) and the features are used to detect dementia. They showed that support vector machine provides better performance than logistic regression. They also determine some key features that contribute more to detecting dementia like, gap before speaking, differences in fundamental frequency, quality of voice etc. They also showed that their system can help health care persons. Moreover, the proposed method can help medical personnel to spot the early sign of dementia. I. Asghar et al. (Asghar, Cang, and Yu 2016) tried to detect dementia in software-based approach. T. Endo et al. (Endo et al. 2017) proposed an approach to determine the eye movement by using the RGB camera usually used in a laptop or in a smartphone. From this eye movement data, they proposed a method to detect dementia. For a proper diagnosis, accurate detection of eye movement is essential. But detection of eye movement is difficult if there is any noisy eye localization. A binary classification of eye movement velocity is used to determine eye movement from the timing when Iris begin and finish moving. A discriminant analysis based classification is used to detect these moments. From the above function, they use two diagnosis system. They use a target mark visible at the different point in a white screen of a computer and track the eye movement. The second one tracks the eye movement when the subject read texts from the web browser. They showed that their method detects iris movement more accurately that is required for dementia detection. M. H. Acharya et al. (Acharya et al. 2016) have tried detect dementia. their proposed idea is to implement a dementia detection system that can run on an Android device. They choose android because, recently android is accepted by a wide variety of smart devices (Tablet, Wristwatch, Cell Phone) as well as it is an open source operating system. They use ”GPS Navigator”, ”Fall Detection System”, ”Mind Games”, ”Doctor-Finder” and ”Emergency” functions in their proposed system to detect dementia. H. Nikamalfard et al. (Nikamalfard et al. 2012) In their research, they describe the monitoring of individuals as well as assessing activity pattern visualization system for early detection of dementia. In their method, they gather several sensor data related to a subject. This sensory data can provide very vital information for assessing the patient activity pattern. Using these data their method can able to detect unusual events and can be used as a very useful impetus for the cognitive test.

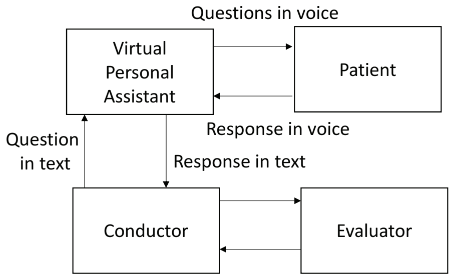


Figure 2: System Components

Proposed Cognitive Test

In our model, we are considering four functional blocks as in Figure 2. First, we have a patient for whom the cognitive test will be conducted. VPA, which facilitates the proposed model by providing text-to-voice and voice-to-text conversion features to interact with the patient. After that the conductor, which will maintain interactions (the question and answer sessions). It will monitor the conversation as well as the response delay of the patient. Finally, the evaluator will verify the patient responses as stated in the previous section. It will compute the level of dementia in HDS-R scale, too.

According to the HDS-R test, a patient is examined by asking nine questions in Figure 1, which basically test the memory recalling and reasoning capability of the patient. In normal situation, we call it ”normal state”, the proposed system just monitoring conversation between a person and a VPA. The system examines each interaction whether it is close to one of the nine HDS-R questions. If the similarity of an interaction is close to a HDS-R question, the system evaluates the interaction based on the HDS-R score. The system monitor the score within a certain period of time. If the score of the time exceeds a certain threshold, the system changes the state into ”warning state”. In warning state, a VPA is forced to ask the nine questions of HDS-R and evaluate answers of the questions. Most of recent VPAs are capable to do such interactions in a natural language user interface, which is physically and mentally comfortable for the person.

Cognitive Test in Normal State

In the normal state, the proposed system monitor the conversation between a person and the VPA. The proposed system then try to find similarity between the conversation and HDS-R. For example, the if the VPA asked for a person age if he gives an answer then the answer consider similar to the first question of HDS-R. We can divide the HDS-R questions into categories like age, orientation in time, orientation in place, repeating ability, subtraction ability, backward counting ability, recalling ability, object recalling, generating common item list.

In this state, the proposed system match the conversation of VPA and a person to the above categories. Considering the following conversation.

```
Person: Find some good hotel near Hiroshima
peace memorial park.
VPA: would you like to provide me your age.
Person: 63 years.
VPA: Houston, Westin Oaks, Whitehall. Please
Give your choice.
Person: Please repeat one more time
VPA: Houston, Westin Oaks, Whitehall. Please
Give your choice.
Person: I can't recall.
```

From the above short conversation we can say here we find a similar situation for HDS-R question-1 and Question-4. So, from the spontaneous conversation when applicable we can apply HDS-R implicitly also able to do some evaluation based on the answers. So, we take each pair of interaction between VPA and person then match it with HDS-R questions, whenever a match found then evaluate the answer of the person.

For finding similar HDS-R like questions from the spontaneous conversation of VPA and a person in this work we use four matrices as described in the work (Lytinen and Tomuro 2002) the matrices are term vector similarity, coverage, semantic similarity and question type similarity. We also normalize each of the matrices in between zero to one.

We compute each of the four metrics but at first, we use POS (Part Of Speech) tagger that allocates a part of speech for each word. The result is then stored as a term vector and a question type. Here, term vector shows the weight of the term found in each question. We find the stemmed word as a term vector in a question. TF-IDF (Term Frequency Inverse Document Frequency) (Salton and McGill 1983) usually use in Information Retrieval(IR). We use TF-IDF to obtain the weight of each term. The weight w_i for each term t_i is computed as

$$w_i = \left(1 + log\left(tf_i\right)\right) \frac{logN}{df_i}.$$

At this point, we want to compare similarity of questions; so, number of questions here consider as N, df_i is the number of question where the term t_i founds, and tf_i is the frequency of the term t_i in the question.

The user interaction with VPA is compare with each HDS-R question, and the four similarity metrices mentioned above are computed. At first, we compute term vector similarity. Each of the HDR-S question we use a term vector as $v_h = \langle w_{h1}, w_{h2}, ..., w_{hn} \rangle$ and each of the VPA question we use a term vector as $v_v = \langle w_{v1}, w_{v2}, ..., w_{vn} \rangle$. The cosine between the two vectors act as the similarity measurement as follows.

$$cos(v_h, v_v) = \frac{\sum w_{hi} w_{vi}}{\sqrt{\sum w_{hi}^2} \sqrt{\sum w_{vi}^2}}$$

We use the percentage of HDR-S question terms that found in a VPA question as a measurement to compute the second metric coverage.

Here, semantic similarity is measured with the help of WordNet, for measuring sematic word matching in VPA question and HDS-R question we calculate minimum path between WordNet concept (synonym sets) with respect to VPA question terms and HDS-R question terms. In general, $\delta\left(t_1, t_2\right)$ the semantic distance between two terms t_1 and t_2, each of which has n and m WordNet senses $S1 = \{s_1, s_2, ..., s_n\}$ and $S2 = \{r_1, r_2, ..., r_m\}$, is the minimum of all possible pair-wise semantic distances between S_1 and S_2, that is, $\delta\left(t_1, t_2\right) = min_{si\epsilon S1, rj\epsilon S2} D\left(s_i, r_j\right)$, where $D\left(s_i, r_j\right)$ is a path length between WordNet synsets s_i and r_j. If there is no path between any of the synsets of t_1 and t_2, then $\delta\left(t_1, t_2\right) = \infty$.

Then, the semantic similarity between the VPA question $5T_u = \{u_1, ..., u_n\}$ and a HDS-R question $T_h = \{h_1, ..., h_m\}$ defined as follows:

$$sem\left(T_u, T_h\right) = \frac{I\left(T_u, T_h\right) + \left(T_h, T_u\right)}{|T_u| + |T_h|}$$

where

$$I\left(T_x, T_y\right) = \sum \frac{1}{1 + min_{y\epsilon T_y} \delta\left(x, y\right)}$$

and $|T_x|$, $|T_y|$ denote the size of T_x and Ty. Thus, $sem\left(T_u, T_h\right)$ is essentially a metric which is the normalized sum of the inverse of pair-wise semantic distances between all words in T_x and T_h measured from both directions.

Lastly, by comparing the question type in VPA and HDS-R, similarity in question type is calculated. A similarity matrix is generated, it usually portrait the amount of nearness between question types.

From this four matrices we find similarity between the VPA and user interaction and HDS-R Question if the similarity is more than 80% then we go for evaluation of the question answer pair of the VPA and user. If the evaluation of the answer is poor then we go for explicit HDS-R test as described in the next section.

Cognitive Test by HDS-R

Here, we use Latent Semantic Analysis (Dumais 2004) to match questions and answers to the 9 questions of the HDS-R test. For each matched question and answer, we evaluate the response as given bellow.

HDS-R test has nine questions and also well defined answers. So, for each question we have to match users answer with the HDS-R answers. How can we judge each of the questions answer is given bellow.

Question 1 During conversation to VPA, when the user is asked for her/his age and the answer is acceptable within the error from correct age is within two-year. We add one point to the score for this question.

Question 2 This question is about current date and day of the week. When we detect a question about current date and day of the week in the conversation, we check the answer and add one point to the score if it is correct.

Question 3 For this question first, we find out the current location by Global Positioning System (GPS) of the device then check a question and the answer of current location also we check the response time for the question. Right answer within 5 seconds give 2 point and right answer after 5 seconds with two option give 1 point for the correct answer.

Question 4 This question is about repeating three common unrelated words. In general, this kind of question does not exist in natural conversation. We have to intentionally include this question, for example, when there is a silence in a conversation like a quiz. Similar to this question, question no. 5 to 9 have to be intentionally included when we find a necessity by observing a problem in the answers for question no. 1 to 3. If it shows exact matching for the three words then the answer is consider as right answer. One point is added to the score for correct matching of each word.

Question 5 Question 5 is about the serial subtraction of 7s from 100. At first, ask the user to subtract 7 from 100. If the answer is wrong then discard the question and go to question no 6. If the first one is right then again ask to subtract 7 from the answer. For correct answer add one point to the score.

Question 6 This question is about backward digit counting so here we have to consider the right answer as well as the order if any wrong answer found then discard the current question and go to next question. When we found that backward counting of 6, 8 and 2 is 2, 8, and 6, then add one point and proceed to other backward digit counting question, i.e., 3-5-2-9, in which correct answer is also one point.

Question 7 It is about recalling three words used in question no 4. We can evaluate it as described in question no 4.

Question 8 In this question, we use five unrelated objects (a key, a cigarette, a watch, a pen and a coin). Which are shown to them as a picture using VPA or spoken by VPA without a display device. After hiding the object ask them to recall. For each correct answers, we add one point to the score.

Question 9 VPA asks to name all vegetables that come to mind and waits untill there is no further answer after 10 seconds from the last one. For each correct name after the 5th will be evaluated as one point.

Conclusion and Future Work

For developing our proposed dementia detection system application, now we are in the preliminary testing stage. We have just checked and verify our system by collecting data as a mimic dementia patient. The speech of old people is different, so we need extra care to recognize their speech. It also needs extra care to deal with the mental situation of old people. Privacy of each individual is also vital, we have to secure and anonymize their sensitive data. All VPA are the proprietary system of the different corporation, so we need proper collaboration with them. As a future work, we will convey a series of experiments on real subject and determine the effectiveness of the model. Moreover, from the voice activity, we believe it will also be possible to examine some other features those could be found in a dementia patient like hesitation, emotional problem and etc.

Acknowledgments

This work is supported by KAKENHI (16K00155,17H01823) Japan. Saleh Ahmed and Mahboob Qaosar are supported by Japanese Government MEXT Scholarship.

References

Abe, Y.; Toya, M.; and Inoue, M. 2013. Early detection system considering types of dementia by behavior sensing. In *IEEE 2nd Global Conference on Consumer Electronics (GCCE)*, 348–349.

Acharya, M. H.; Gokani, T. B.; Chauhan, K. N.; and Pandya, B. P. 2016. Android application for dementia patient. In *International Conference on Inventive Computation Technologies (ICICT)*, volume 1, 1–4.

Alkabawi, E. M.; Hilal, A. R.; and Basir, O. A. 2017. Computer-aided classification of multi-types of dementia via convolutional neural networks. In *IEEE International Symposium on Medical Measurements and Applications (MeMeA)*, 45–50.

Asghar, I.; Cang, S.; and Yu, H. 2016. Software based assistive technologies for people with dementia: Current achievements and future trends. In *International Conference on Software, Knowledge, Information Management Applications (SKIMA)*, 162–168.

Dumais, S. T. 2004. Latent semantic analysis. *Annual Review of Information Science and Technology* 38(1):188–230.

Endo, T.; Ukita, N.; Tanaka, H.; Hagita, N.; Nakamura, S.; Adachi, H.; Ikeda, M.; Kazui, H.; and Kudo, T. 2017. Initial response time measurement in eye movement for dementia screening test. In *IAPR International Conference on Machine Vision Applications (MVA)*, 262–265.

Imai, Y., and Hasegawa, K. 1994. The revised hasegawa's dementia scale (HDS-R) - evaluation of its usefulness as a screening test for dementia. *Hong Kong Journal of Psychiatry* 4:20–24.

Jimison, J. B.; Pavel, M.; Pavel, J.; and McKanna, J. 2004. Home monitoring of computer interactions for the early detection of dementia. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, 4533–4536.

Lytinen, S. L., and Tomuro, N. 2002. The use of question types to match questions in faqfinder.

Nikamalfard, H.; Zheng, H.; Wang, H.; Jeffers, P.; Mulvenna, M.; McCullagh, P.; Martin, S.; Wallace, J.; Augusto, J.; Carswell, W.; Taylor, B.; and McSorley, K. 2012. Knowledge discovery from activity monitoring to support independent living of people with early dementia. In *IEEE-EMBS International Conference on Biomedical and Health Informatics*, 910–913.

Salton, G., and McGill. 1983. *Introduction to Modern Information Retrieval*. Boca Raton: McGraw-Hill College.

Shigemori, T.; Kawanaka, H.; Hicks, Y.; Setchi, R.; Takase, H.; and Tsuruoka, S. 2016. Dementia detection using weighted direction index histograms and svm for clock drawing test. *Procedia Computer Science* 96(Supplement C):1240–1248. Knowledge-Based and Intelligent Information and Engineering Systems: Proceedings of the 20th International Conference KES-2016.

Tanaka, H.; Adachi, H.; Ukita, N.; Ikeda, M.; Kazui, H.; Kudo, T.; and Nakamura, S. 2017. Detecting dementia through interactive computer avatars. *IEEE Journal of Translational Engineering in Health and Medicine* 5:1–11.

Measuring Cognitive Bias in Spoken Interaction and Conversation: Generating Visual Representations

Christina Alexandris
National and Kapodistrian University of Athens
calexandris@gs.uoa.gr

Abstract

The present approach targets to assist decision-making by identifying and by-passing Cognitive Bias of speakers-participants and evaluators in regard to spoken texts.

Registering Spoken Interaction

Registered spoken interaction is integrated and processed in a database under development for determining and evaluating Cognitive Bias in spoken journalistic texts. The target is to facilitate evaluation of spoken interviews (including short on-line interviews via Skype) and discussions in the Media and to assist decision-making by identifying and by-passing Cognitive Bias of speakers-participants and evaluators.

From this perspective, databases based primarily on task-specific dialogs (Tung et al., 2013) and evaluation strategies for collaborative dialogs (Yang et al, 2012, Wang 2013) are not fully compatible with the conversation and interaction type concerned, where there is expression of sentiment, statement of opinion or even persuasion. In these texts, the discourse structure may either be compatible to turn-taking in "push-to-talk conversations" (strict protocol in managing the interview or discussion and turn-taking) (Taboada, 2006) or compatible to models where each participant selects self (Wilson, 2005, Sacks et al., 1974). The approach is based on data and observations provided by professional journalists. Collected data involves transcriptions from two-party or multiple party discussions of spoken journalistic texts (Program M.A in Quality Journalism and Digital Technologies, Danube University at Krems, Austria in collaboration with the Institution of Promotion of Journalism Ath.Vas. Botsi, Athens) (52 transcribed interviews by 30 journalists on domestic and foreign policy). The database, currently in the stage of development, combines practices from dialog systems with features from designed applications for journalists (Alexandris et al., 2015).

Path generation of the interaction is modeled and implemented based on user interactions registered in spoken dialog systems, in the domains of consumer complaints and mobile phone services call centers (Nottas et al., 2007, Floros and Mourouzidis, 2016). The System generates a visual representation from the user's interaction, tracking the corresponding selected keywords in the dialog flow. The same model is applied for tracking topics and generating models in transcribed spoken journalistic texts. Specifically, there is an interactive generation of registered paths, similar to paths with generated sequences of recognized keywords (Nottas et al., 2007, Floros and Mourouzidis, 2016). Thus, a keyword (topic) may be repeated (Repetition) or related to a more general concept (or global variable) (Lewis, 2009) (Generalization) or related to keywords (topics) concerning similar functions (Association). A keyword involving a new command or function is registered as a new topic (New Topic). The "path" of interaction is generated with the sequence of topics chosen by the user and the perceived relations between them, forming distinctive visual representations according to its content.

Cognitive Bias can be determined from the form of generated visual representations of dialog flow, for evaluating success or failure of spoken interaction (I). This is related to by-passing Confidence Bias of speakers-participants and evaluators (Hilbert, 2012). Cognitive Bias is also measured in the form of triple tuples as perceived relations-distances between word-topics (II), related to a type of Lexical Bias concerning semantic perception (Trofimova, 2014).

Generating Visual Representations

Topics are defined at a local level with the activation of the "Identify Topic" command, in respect to the question asked or issue addressed by the interviewer or moderator.

This feature, based on previous research concerning the interactive annotation of pragmatic features in transcribed journalistic texts (Alexandris et al., 2015), allows the content of answers, responses and reactions to be checked in respect to the question asked or issue addressed. Topics, treated as local variables, are registered and tracked. The automatic signalization of nouns by the Stanford POS Tagger in each turn taken by the speakers in the respective segment in the dialog structure provides assistance in choice of topic.

With the activation of the "Identify Relation" command, relation types between topics are determined by the user. In the domain of journalistic texts, these relations cannot be strictly semantic: automatic processes may result to errors. The user choses the type of relation ("Repetition", "Association", "Generalization" or "Topic Switch") between the topic of the question or issue addressed with the topic of the respective response or reaction (Alexandris et al., 2015). The "Repetition" relation ("REP" tag) involves the repetition of the same word or synonym and corresponds to the generation of the shortest distance between defined topics ("Distance 1"- one dash in generated pattern). The "Association" relation ("ASOC" tag, "Distance 2"), defined by the user's world knowledge (can be evaluated with a lexicon or Wordnet) is represented as a longer line to the next word-node (two dashes). The "Generalization" relation ("GEN" tag), also defined by the user's world knowledge (comparable to a lexicon or Wordnet) corresponds to the generation of the longest distance between defined topics ("Distance 3"-three dashes). The "Topic Switch" relation ("SWITCH" tag) is used when the topic of a discussion or interview changes between selected topics without any evident semantic relations. "Topic Switch" (Distance 4: slash "/") generates a new line - a break, in the sequence of topics. Examples of segments in (interactively) generated patterns from user-specific choices between topics (Tpc) are the following:

- "Britain"-"the UK" (REP-1):TpcA-TpcB.
- "propaganda"-"social-media"(ASOC-2):TpcA--TpcB.
- "police"-"security"(GEN-3):TpcA---TpcB.
- "security"/"entrepreneurship"(SWITCH-4):TpcA/ TpcB.

The distances (II) between topics in the generated patterns (I) are registered as triple tuples (triplets): (Britain, the UK, 1), (propaganda, social media, 2), (police, security, 3), (security, entrepreneurship, 4).

Measuring and Evaluating Cognitive Bias

Content (i) and form of generated patterns (for example, multiple breaks) (ii) as visual representations of Cognitive Bias target to depict:

- (1) Degree in which all topics are addressed.
- (2) What topics are avoided – either by changing a topic or by persisting to address the same topic: Observed to be evident in length and form of generated pattern.
- (3) How participants may be lead or even forced into addressing a topic –by association or generalization: This is also observed in length and form of generated patterns.

Therefore, targeting to by-pass Confidence Bias (Hilbert, 2012) of users-participants and evaluators (II), the above-presented points allow the determination of the participants in the conversation (or interview) who were successful in their spoken interaction and the participants who were less successful. Simultaneously, the perceived relations-distances between word-topics perceived by the user, related to the above-stated type of Lexical Bias, (Trofimova, 2014) are generated and measured in the form of triple tuples (II). The generated patterns contribute to a user-independent evaluation of spoken Human-Human conversation and interaction, similarly to user-independent evaluation of spoken dialog systems (Williams et al., 2017), where speed and correctness are of crucial importance (Lewis, 2009). Varying degrees of user's familiarity with dialog systems or user-friendly interfaces in spoken interaction result to different perceptions of successful interactions and may "forgive" occasional errors (Nass and Brave, 2005, Cohen, 1997): Errors in spoken input or a longer duration of interaction due to complications in the dialog may not always correspond to negative evaluation. Similarly, varying degrees of familiarity and bias with topics discussed in spoken journalistic texts result to different perceptions of successful conversations or debates and may "forgive" any complications or mistakes.

We also note that data from transcriptions and respective visual representations created so far indicates cases of observed differences between identified topic relations among some journalists that are non-native speakers of English (especially in respect to "ASOC" and "SWITCH"). Differences may in some cases be attributed to lack of world knowledge of the language community concerned (Paltridge, 2012, Hatim, 1997, Wardhaugh, 1992), particularly in non-native speakers. This implies that the international public may often perceive and receive different and/or incomplete information in respect to evaluating conversation and interaction (Yu et al., 2010, Alexandris, 2010, Ma, 2010, Pan, 2000). Topics and words generating diverse reactions and choices from users result to the generation of different forms of generated visual representations for the same conversation or interaction:

- "Country Z" –"defense spending" (ASOC) or (SWITCH)

Further Research

Targeted enrichment with more registered interactions - possibly in a graphic form similar to discourse trees (Marcu, 1999, Carlson et al., 2001), will provide more concrete results and data for statistical analysis, possibly contributing to the evaluation of a user's familiarity, perception and world knowledge in other domains and applications such as education-training and virtual negotiators.

References

Alexandris, C., Nottas, M., Cambourakis, G. 2015. Interactive Evaluation of Pragmatic Features in Spoken Journalistic Texts. M. Kurosu ed. *Lecture Notes in Computer Science*, 9171: 259-268.

Alexandris, C. 2010. English, German and the International "Semi-professional" Translator: A Morphological Approach to Implied Connotative Features. *Journal of Language and Translation*, Vol. 11, 2, Sejong University, Korea: 7- 46.

Carlson, L., Marcu, D., Okurowski, M. E. 2001. Building a Discourse-Tagged Corpus in the Framework of Rhetorical Structure Theory. In *Proceedings of the 2nd SIGDIAL Workshop on Discourse and Dialogue*, Eurospeech 2001, Denmark.

Cohen, P., Johnston, M., McGee, D., Oviatt, S., Pittman, J., Smith, I., Chen, L., Clow, J. 1997. Quickset: Multimodal Interaction for Distributed Applications. In *Proceedings of the 5th ACM International Multimedia Conference*, 31-40.

Du, J., Alexandris, C., Mourouzidis, D., Floros, V.,Iliakis, A. 2017. Controlling Interaction in Multilingual Conversation Revisited: A Perspective for Services and Interviews in Mandarin Chinese. M. Kurosu ed. *Lecture Notes in Computer Science* 10271: 573–583.

Floros, V., Mourouzidis, D. 2016. Multiple Task Management in a Dialog System for Call Centers. Master's Thesis, Department of Informatics and Telecommunications, National University of Athens, Greece.

Hatim, B. 1997. *Communication Across Cultures: Translation Theory and Contrastive Text Linguistics*. Exeter: University of Exeter Press.

Hilbert, M. 2012. Toward a Synthesis of Cognitive Biases: How Noisy Information Processing Can Bias Human Decision Making. *Psychological Bulletin*, Vol 138(2), Mar 2012: 211-237.

Lewis, J.R. 2009. Introduction to Practical Speech User Interface Design for Interactive Voice Response Applications, IBM Software Group, USA, Tutorial T09 presented at HCI 2009 San Diego, CA, USA.

Ma, J. 2010. A comparative analysis of the ambiguity resolution of two English-Chinese MT approaches: RBMT and SMT. *Dalian University of Technology Journal*, 31(3): 114-119.

Marcu, D. 1999. Discourse trees are good indicators of importance in text. I.Mani and M. Maybury eds. *Advances in Automatic Text Summarization*, Cambridge, MA: The MIT Press, 123-136.

Nass, C. and Brave, S. 2005. *Wired for Speech: How Voice Activates and Advances the Human-Computer Relationship.* Cambridge, MA: MIT PRESS.

Nottas, M., Alexandris, C, Tsopanoglou, A. Bakamidis, S. 2007. A Hybrid Approach to Dialog Input in the CitzenShield Dialog System for Consumer Complaints. In *Proceedings of HCII 2007*, Beijing, Peoples Republic of China, Heidelberg: Springer.

Paltridge, B. 2012. *Discourse Analysis: An Introduction*. London: Bloomsbury Publishing.

Pan, Y. 2000. Politeness in Chinese Face-to-Face Interaction. *Advances in Discourse Processes Series* Vol. 67. Stamford, CT, USA: Ablex Publishing Corporation.

Sacks, H., Schegloff, E. A., Jefferson, G. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, Vol. 50: 696-735.

Taboada, M. 2006. Spontaneous and non-spontaneous turn-taking. *Pragmatics*, Vol.16 (2-3): 329-360.

Trofimova, I. 2014. Observer Bias: An Interaction of Temperament Traits with Biases in the Semantic Perception of Lexical Material. *PLoS ONE* 9(1): e85677.

Tung, T., Gomez, R., Kawahara, T., Matsuyama, T. 2013. Multiparty Human-Machine Interaction Using a Smart Multimodal Digital Signage. *Lecture Notes in Computer Science*, 8007, 2013: 408-415.

Wang, H., Gailliot, A., Hyden, D., Lietzenmayer, R. 201). A Knowledge Elicitation Study for Collaborative Dialogue Strategies Used to Handle Uncertainties in Speech Communication While Using GIS. In M.Kurosu ed.*Lecture Notes in Computer Science* 8007: 135-146.

Wardhaugh, R. 1992. *An Introduction to Sociolinguistics*, 2nd edition. Oxford, UK: Blackwell.

Williams, J.D., Asadi, K., Zweig, G. 2017. Hybrid Code Networks: practical and efficient end-to-end dialog control with supervised and reinforcement learning. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada: 665–677.

Wilson, K. E. 2005. An oscillator model of the timing of turn-taking. *Psychonomic Bulletin and Review* 2005:12 (6). 957-968.

Yang, Z., Levow G.A., Meng, H. 2012. Predicting User Satisfaction in Spoken Dialog System Evaluation With Collaborative Filtering. *IEEE Journal of Selected Topics in Signal Processing*, Vol. 6, Issue: 8, Dec. 2012: 971 – 981.

Yu, Z., Yu, Z., Aoyama, H., Ozeki, M., Nakamura, Y. 2010. Capture, Recognition, and Visualization of Human Semantic Interactions in Meetings. In *Proceedings of PerCom*, Mannheim, Germany, 2010.

A Study on the UI of Musical Performance System and Score Representation

Sachiko Deguchi
Faculty of Engineering, Kindai University, Hiroshima, Japan
deguchi@hiro.kindai.ac.jp

Abstract

This paper describes the development and evaluation of the UIs and Scores of musical performance system. The aim of this research is to provide a musical tool for elderly people and caregivers. The UIs are designed on tablet PCs, which look like keyboards. Five UIs are evaluated: plain keyboard, and keyboards with note names, numbers, colors and shapes. A staff notation score was used for the plain keyboard, and four types of scores represented by note names, numbers, colors and shapes were used for other UIs. The result of the experiment indicates that the UIs and scores of note name representation and number representation would be useful to play for people who are not familiar with staff notation and that those of number representation would be useful to play and sing at the same time. The result also indicates that the UI and scores of color representation could be used for some people who have difficulty reading numbers. It is also indicated that people in their 60s and 70s can use the UI and scores of number representation.

Introduction

The aim of this study is to provide a musical tool to improve the quality of life of elderly people. This tool can also be a communication tool for elderly people and caregivers. It is known that musical memory can be preserved even in Alzheimer's disease, and the evidence has been indicated by using fMRI (Jacobsen 2015).

In the field of music therapy, music listening and/or singing are commonly used (Groene 1993) (Raglio, et al. 2008) (Satoh, et al. 2015). However, it is difficult to use musical instrument for elderly people who have little experience with musical performance. There are two problems: manipulating a musical instrument may be difficult, and staff notation scores might be the barrier to musical performance. In this research, several User Interfaces (UIs) for musical performance and several musical scores represented for these UIs have been developed and evaluated.

There are other approaches to the musical performance system. Many new systems have been proposed and evaluated in the field of Computer Music (Zbyszynski, et al. 2007) (Hochenbaum, et al. 2010) (Oh, et al. 2010) (Brown, Nash, and Mitchell 2017), however, these systems are used for improvisation or used based on staff notation. On the other hand, a new instrument, e.g. Veeh-Harp was developed along with new score representation (Veeh 1987), and the concept of this instrument can be applied to some support system for performance. However, the aim of our research is to provide a system which can be used to play and sing by reading musical scores.

The UIs and Scores

A system with several UIs for musical performance has been developed in this research, which works on Windows tablet PCs. Musical scores of several representations are provided for these UIs.

UIs of Musical Performance System

This system has several user interfaces as follows: (1) UI of layered keyboard (UI-1), (2) UI with note names on the keyboard (UI-2), (3) UI with numbers on the keyboard (UI-3) as Figure 1, (4) UI with colors on the keyboard (UI-4) as figure 2, (5) UI of geometric shaped keys (UI-5) as figure 3, and (6) Different layout of UI-3 (UI-6).

The layout of notes are the same in five UIs (UI-6 is different). C3 to B3 keys are in the bottom row, sharp/flat keys are in the next row, C4 to B4 keys are in the next row, and so on. Figure 1 shows UI-3 where numbers are put on the keys: 1 is on C4 key, 2 is on D4 key, and 7 is on B4 key. A number with an upper dot means the pitch is one octave higher than the number, and a number with a lower dot means the pitch is one octave lower than the number.

This system provides a sound source of piano (C2 to C7) and organ (C2 to C7) in WAVE data format. These sounds were generated by additive synthesis.

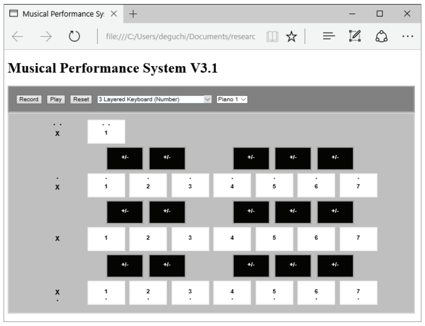

Musical Performance Sy
file:///C:/Users/deguchi/Documents/researc
Musical Performance System V3.1
Record
Play
Reset
3 Layered Keyboard (Number)
Piano 1
x
1
+/-
1
2
3
4
5
6
7

Figure 1: UI with Numbers

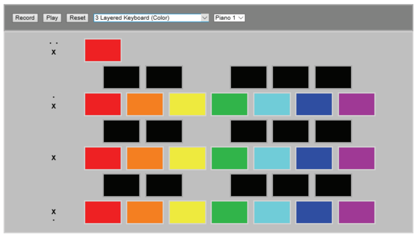


Figure 2: UI with Colors

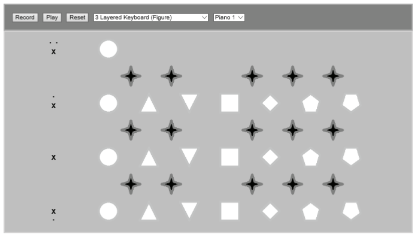


Figure 3: UI of Shaped Keys

Users of this system can use piano touch or organ touch according to the sound source. In piano touch mode, the sound starts when a key is touched and it decays like a piano. In organ touch mode, the sound starts when a key is touched and it ends when the finger is released. Users can also use multi-touch so that they can play chords.

This system can read a musical score from score database which was developed in this research, and play music so that a user can listen to the melody before performance. This system can also record user's performance to support practice and improvisation.

This system was implemented by HTML, JavaScript and WAVE data, and it is working without perceptible time delay on Windows tablet PCs (HP Pro Tablet and NEC LAVIE Tab). We have not used the tools for building UIs (Bryan, et al. 2010). HTML and JavaScript are used because of the flexibility of building UIs, and WAVE data is used because of the response speed. Before developing this system, we developed a system using C++ and MIDI sound but the system worked with time delay.

Score Representaion

Each user interface needs different score form: (1) staff notation for UI-1, (2) note name representation for UI-2, (3) number representation for UI-3 and UI-6, (4) color representation for UI-4 and (5) shape representation for UI-5.

The scores used for the experiments were notated manually on Excel sheets because the score editor had not been developed. This notation is based on the scores used by Ikuta school of Koto music (Miyagi 1969) which is one of Japanese traditional music. Scores of Koto music were notated simply in Edo period, and they were described precisely in Meiji period (19th century). The duration of a note is represented as the width of the space where the note is notated. Figure 4 shows the example. The notation of Ikuta scores is vertical direction, however this notation is horizontal direction.

The score editor is now under development (Deguchi 2017). The scores are generated from musical score database which was developed in this study. This representation is based on the scores used by Yamada school of Koto music (Nakanoshima 1954), and this notation is similar to the scores of Taisho Koto which was developed in the beginning of 20th century. A single line is added to the 8th note, and a double line is added to the 16th note, and so on. Figure 5 shows the example.

There is an editor which can display Ikuta scores, Taisho Koto scores and other scores of Japanese traditional music (Tanishi 2010), however it cannot be used for new representations. The editor of our research can display scores in different representations: scores of note names, numbers, colors and shapes.

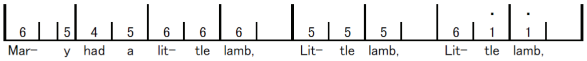


Figure 4: Score using Numbers for the Experiments

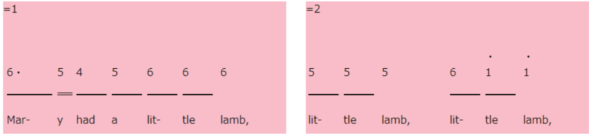


Figure 5: Score using Numbers Generated by the System

Score database was developed using public domain musical pieces (the voice part of one piece is copyrighted). The pieces are notated in Humdrum format (Huron 1998) and saved in text files so that users can easily edit the files. The scores in this DB have been input by (1) editing Humdrum files directly, or (2) reading sheet music by OCR software (KAWAI ScoreMaker) and transforming MusicXML format into Humdrum format (Sapp 2004). The contents of the DB are as follows.

Japanese songs: 17 songs such as Sakura.
English songs: 10 songs such as Mary had a little lamb.
Classical music: 3 pieces such as Air on the G String.

Experiments of UIs and Scores

Two experiments to evaluate these UIs and scores are described in this section.

Methods

The first experiment was done to compare UI-1, UI-2 and UI-3 in Feb. 2016 (Deguchi 2016). The conditions of the experiment are as follows.
Examinees: 16 students of Engineering Department who were not familiar with keyboard instruments.
Songs used for the experiment: Sakura for UI-1, Harunoogawa for UI-2, Yuyake-koyake for UI-3.

The second experiment was carried out to compare UI-3, UI-4 and UI-5 in Feb. 2017.
Examinees: 16 students of Engineering Department who were not familiar with keyboard instruments (8 students are the same students in the first experiment).
Songs used for the experiment: Sakura for UI-3, Harunoogawa for UI-4, Yuyake-koyake for UI-5.

The following conditions are the same in both experiments.
Condition before the experiment: The examinees did not practice the system.
Condition during the experiment: First, the examinees played the system using each UI and score. Next, the examinees played and sang using each UI and score.

The examinees answered the questions by rating 4, 3, 2 or 1 (4:positive, 3:mildly positive, 2:mildly negative, 1:negative) for each UI and its score after using them. Questions are as follows.
Q1: Is the score easy?
Q2: Is the UI easy?
Q3: Is it easy to play?
Q4: Is it easy to play and sing at the same time?

Results and Discussion

The results of the experiments are as follows. The mean value of each question for each UI and its score is shown in Table 1 and Table 3. Table 1 indicates that UI-2 and its score (note name representation) and UI-3 and its score (number representation) might be easier than UI-1 and its score (staff notation) for the people who were not familiar with musical performance. Table 3 indicates that UI-3 and its score and UI-4 and its score (color representation) may be easier than UI-5 and its score (shape representation).

Paired sample t-test was used for the comparison of the mean values of each question for two UIs. UI-1 and UI-2, UI-1 and UI-3, and UI-2 and UI-3 are compared in the first experiment. The degrees of freedom is 15, and the critical value for significance level of 0.05 (two-tailed test) is 2.131 and that of 0.01 is 2.947. T-ratio of each comparison is shown in Table 2. Table 2 indicates that UI-2 and UI-3 are easier than UI-1 to play the system. It also indicates that UI-3 is easier than UI-1 and UI-2 to play and sing at the same time. Note names would conflict with songs when a user uses UI-2 and its score. This result does not contradict the fact that some of Japanese traditional music which has instrument part and voice part use numbers for score notation.

UI-3 and UI-4, UI-3 and UI-5, and UI-4 and UI-5 are compared using paired sample t-test in the second experiment. The degrees of freedom and the critical values are the same as described above. T-ratio of each comparison is shown in Table 4. Table 4 indicates that UI-5 is difficult to use. It also indicates that the score for UI-3 is easier than the score for UI-4, however, it cannot indicate that UI-3 is easier than UI-4 to play or to play and sing. UI-4 and its score (color representation) could be used for some people who have difficulty reading numbers.

Table 1: Mean Values of Evaluation of UI-1, UI-2 and UI-3

	UI-1	UI-2	UI-3
Q1	2.75	3.38	3.44
Q2	2.63	3.25	3.06
Q3	2.19	3.25	3.19
Q4	2.38	2.56	3.19

Table 2: T-ratios of T-test for UI-1, UI-2 and UI-3

	UI-1 vs. UI-2	UI-1 vs. UI-3	UI-2 vs. UI-3
Q1	-1.91	-3.15	-0.32
Q2	-3.10	-1.82	1.38
Q3	-7.41	-5.48	0.29
Q4	-1.14	-3.90	-3.48

Table 3: Mean Values of Evaluation of UI-3, UI-4 and UI-5

	UI-3	UI-4	UI-5
Q1	3.44	2.56	1.88
Q2	3.50	3.00	2.13
Q3	3.19	2.69	1.88
Q4	3.25	2.63	2.19

Table 4: T-ratios of T-test for UI-3, UI-4 and UI-5

	UI-3 vs. UI-4	UI-3 vs. UI-5	UI-4 vs. UI-5
Q1	3.95	7.68	3.91
Q2	2.24	6.21	3.66
Q3	1.52	6.01	3.31
Q4	1.99	4.58	2.41

A Course for Elderly People

This system was used in an extension course of Faculty of Engineering, Kindai University, in Nov. 2017 as follows.

Participants: 24 adults (40s: 2, 50s: 2, 60s: 12, 70s: 7, Unknown: 1).

UI: UI-3 (number representation) was used. The notes arranged on UI-3 are C3 to C6, however UI-3 of higher notes (C4 to C7) and UI-3 of lower notes (C2 to C5) were also provided.

Scores: Number representation scores of 13 Japanese songs, 4 English songs and 3 pieces of Classical music were used.

The lecturer explained how to read scores and how to play the system, and then explained about Japanese songs, English songs and Classical music which were used in the course. Total time of explanation was about 30 minutes. The participants practiced Japanese songs, English songs and Classical music using each tablet for about 70 minutes.

Participants answered the questions by rating 4, 3, 2 or 1 for UI-3 and its scores of Japanese songs after this course. The questions are the same as described above. The mean value of each question answered by 60s and 70s is as follows. Q1: 3.84, Q2: 3.56, Q3: 3.74, Q4: 3.05

This result indicates that elderly people can use this system. It also indicates that it might be relatively difficult for elderly people to play and sing at the same time.

Conclusion and Future Work

In this research, four types of UIs and scores were proposed, where notes were represented by note names, numbers, colors or shapes. These UIs with scores and normal UI (plain keyboard) with score (staff notation) were evaluated by experiments. The result indicates that the UIs and scores of note name representation and number representation would be useful to play for people who are not familiar with staff notation and that those of number representation would be useful to play and sing at the same time. This study also indicates that the UI and scores of color representation could be used for some people who have difficulty reading numbers. The result of the extension course indicates that people in their 60s and 70s can use the UI and scores of number representation.

Future work includes research on score DB. There are two versions for score representation: scores as Figure 4 and scores as Figure 5. A new function has been added to the system so that it can generate both types of representation. It is also important how to increase musical pieces in DB. It would be necessary to discuss with the research organizations which have score DB (Hewlett and Selfridge-Field 2001). We are also adding new functions which can reuse the public domain musical pieces by changing rhythms and synthesizing phrases (Deguchi 2017).

Acknowledgments

The author would like to thank H. Suzuki, K. Tanaka, M. Koizumi and J. Nisougi for their contribution to the system development.

References

Brown, D.; Nash, C.; and Mitchell, T. 2017. A User Experience Review of Music Interaction Evaluations. In *Proceedings of the International Conference on NIME*, 370-375. NIME.

Bryan, N.J. et al. 2010. MoMu : a Mobile Music Toolkit. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 174-177. NIME.

Deguchi, S. 2016. Basic Research on User Interface of Music Performance System. In *Proceedings of the Research Meeting 2016 Spring of Japanese Society for Music Perception and Cognition*, 87-90. Tokyo: JSMPC.

Deguchi, S. 2017. Research on Easy Musical Performance System and Score Database. In *Proceedings of the 2017 IEICE General Conference*, IS-1, 86. Tokyo: IEICE.

Groene, R.W. 1993. Effectiveness of Music Therapy 1:1 Intervention with Individuals Having Senile Dementia of the Alzheimer's Type. *Journal of Music Therapy* 30(3): 138-157.

Hewlett, W.B. and Selfridge-Field, E. eds. 2001. *The Virtual Score. Computing in Musicology* (12). Cambridge, Mass: The MIT Press.

Hochenbaum, J. et al. 2010. Designing Expressive Musical Interfaces for Tabletop Surfaces. In *Proceedings of the International Conference on NIME*, 315-318. NIME.

Huron, D. 1998. *Humdrum User's Guide*.
http://www.musiccog.ohio-state.edu/Humdrum/guide.toc.html

Jacobsen, J.H. et al. 2015. Why Musical Memory can be Preserved in Advanced Alzheimer's Disease. *Brain* 138(8): 2438-2450.

Miyagi, M. 1969. *Rokudan no Shirabe. Ikuta School Koto Music.* Tokyo: Hogakusha.

Nakanoshima, K. 1954. *Rokudan no Shirabe. Yamada School Koto Score.* Tokyo: Hogakusha.

Oh, J. et al. 2010. Evolving the Mobile Phone Orchestra. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 82-87. NIME.

Raglio, A. et al. 2008. Efficacy of Music Therapy in the Treatment of Behavioral and Psychiatric Symptoms of Dementia. *Alzheimer Disease and Associated Disorders* 22(2): 158-162.

Sapp, C. 2004. *xml2hum*.
http://extras.humdrum.org/man/xml2hum/

Satoh, M. et al. 2015. Music Therapy Using Singing Training Improves Psychomotor Speed in Patients with Alzheimer's Disease: A Neuropsychological and fMRI Study. *Dementia and Geriatric Cognitive Disorders Extra* 5(3): 296-308.

Tanishi, Z. 2010. *Wagaku Hitosuji*.
https://jonkara.net/soft/wagaku/

Veeh, H. 1987. Veeh-Harfe. http://www.veeh-harfe.de/

Zbyszynski, M. et al. 2007. Ten Years of Tablet Musical Interfaces at CNMAT. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, 100-105. NIME.

A Dynamic Learning Model for a Better Personalized Healthcare Using Mobile Health Tools

Amy Wenxuan Ding
Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213
dingcmu@gmail.com

Abstract

Scientists estimate nearly half of the world's adult population will be overweight or obese by 2030. Widely used mobile devices can provide inexpensive tools to reinforce self-monitoring of weight management behaviors and have a great potential in obesity treatment. However, their effectiveness depends on whether users *actively* responds to suggestions or health interventions displayed. This paper proposes a novel theory-based dynamic learning model to examine how a user's unobserved mind states of activated engagement in weight loss affect her weight management activities. Based on a mobile health app dataset, we find that there exist two mind states (activated vs. inactivated) among the app users. Users in the activated state of weight loss engagement significantly increase their daily steps taken by 57.82% compared to those in the inactivated state when following the health interventions in the app. Further, a simple home-screen reminder of checking the health suggestions in the app targeting inactivated-state users will increase their probabilities and time duration of moving into the activated state by 29% and 38.9%, respectively. As a result, user mind state-based personalized healthcare interventions in the mobile app are shown to be quite effective.

Introduction

Obesity is a major contributor to many chronic diseases including type 2 diabetes, cardiovascular disease, many cancers, and numerous other diseases and conditions. The World Health Organization reports that 1.4 billion adults globally exceed healthy body weight (WHO 2013). This rate increases every year and by 2030 nearly half of the world's adult population will be overweight or obese. In U.S., the projected cost of treating preventable obesity-related diseases is expected to raise by $48-66 billion/year (Wang et al 2011). Widely used mobile devices (e.g., wearables and apps) and the rapid development of artificial intelligence technologies can provide inexpensive tools to reinforce self-monitoring of weight management behaviors or motivate users to adhere to treatment protocols, and offer real-time tailored interventions to improve health outcomes, while reducing the cost and increasing the convenience of treatment delivery and dissemination. However, to achieve this requires a better understanding of individual users' mind states of activated engagement in weight loss and their corresponding impacts on weight management.

Using a mobile app to provide users health interventions for weight loss actually consists of an intertwined cycle of users' offline-online-offline behaviors, reflecting the social cognitive construct of reciprocal causation. Users' online behaviors include browsing the app to look at their physical/dietary activity information, suggestions to reduce weight, etc. Their offline behaviors include physical activities like actual exercise, eating, and seeking coaching. In reality, some users could be in an activated state of weight loss engagement and actively respond to health interventions while others may be casual users in an inactivated state not motivated to follow the suggestions closely. Also a user could change from the inactivated state to activated state from time to time or vice versa. In any circumstance, if a user does not respond to health suggestions in the app by adjusting her *offline* physical activities, the effectiveness of health interventions using the app will be low, and the desired weight loss goal may not be achieved.

Prior research in the field by using artificial intelligence and machine learning approaches has primarily focused on how to offer customized health/wellness recommendations based on sensor recorded data. Combining with users' profiles and the recorded physical activities in terms of calorie consumption, the app provides *mass* customization and suggests wellness information that the app believes is good to each individual without considering their mind states. We know that human mind determines behavior, and users could change their mind states over time in response to stimuli in the app. This dynamic property of mind states is often ignored by conventional learning methods used in mobile health tools (Bacigalupo et al. 2013, Free and

Philips 2013, Williams and French 2011, Manzoni et al. 2011). Also, it is unclear if the displayed health intervention does stimulate users to take offline physical actions.

To fill the gap, this paper presents a novel theory-based dynamic learning model that observes individual user's online tap behavior to automatically learn and identify her unobserved real-time mind state for weight loss. We address the following research questions: (1) How does app-delivered messaging influence user mind states and their dynamics? And (2) Does the mind state of weight loss engagement moderate how effectively health interventions impact users' physical activities?

The Approach and Results

According to the Stimulus-Organism-Response (S→O→R) framework in the human mind theory (Ajzen 1991), information on the app as stimuli S affects users' unobserved mind states O, which then influences their offline physical activities R. We develop a hierarchical Bayes learning model with a first-order hidden Markov chain that performs a backward reasoning from online observations R to infer unobserved O, and the model performs a simultaneous analysis to find optimal mind state-based intervention.

Based on a mobile health app dataset with 250 overweight users for 3 months, we focus on their daily steps taken. The app displays the user's total and average steps up to pervious day, the recommended health intervention (e.g., 20% increase in the average daily steps by far), whether the user follows the intervention, the steps taken on the current day, pages viewed in the app, if the app visit is on a weekend, and the user's demographic information including age and gender.

The results show that there exist two mind states (activated vs. inactivated) among the app users. The simple home-screen reminder message targeting inactivated-state users is quite effective, drastically increasing the user's probability of being in the activated state from 33% to 62% with a significant 29% increase. Also, on average it reduce the user's stay in the inactivated state from 2.70 days to 1.75 days while increasing her time in the activated state from 4 days to 5.56 days (a 38.9% increase).

Interestingly, we find that users' mind states significantly moderate the impact of health intervention on the daily steps taken. As shown in Figure 1, when not accounting for user mind states, a user takes 3660 and 4680 steps daily when not following and following the health interventions, respectively. However, when user mind states are considered, activated-state users take dramatically more steps (3782.7 and 5586.9 steps in the two above cases) than those in the inactivated state (3300.0 and 3540.0 steps). The most noticeable difference between these two groups exists in the case of user following the intervention suggestion (3540.0 vs. 5586.9 steps with a 57.8% difference).

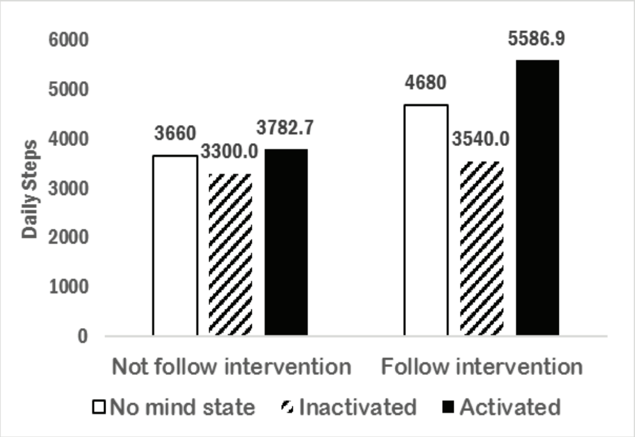


Figure 1. Impacts of Users' Mind States

Overall, the results suggest that it is important to incorporate users' unobserved mind states of weight loss engagement which significantly moderate the impact of health intervention on users' daily steps taken. A simple home-screen reminder of checking the health suggestions in the app targeting inactivated-state users is quite effective to prompt the user to move into the activated state. Therefore, we can design an optimal user mind state-based personalized healthcare intervention in the mobile app based on the dynamics of user mind states in real time, which is proven to be an effective strategy to improve user weight loss management.

References

Ajzen, I. 1991. The Theory of Planned Behavior. *Organization Behavior and Human Decision Processes,* 50: 179-211

Bacigalupo, R., Cudd, P., Littlewood, C. et al., 2013. Interventions Employing Mobile Technology for Overweight and Obesity: An Early Systematic Review of Randomized Controlled Trials. *Obesity Review*, 14(4): 279-291.

Free, C., and Phillips, G. 2013. The Effectiveness of Mobile-health Technologies to Improve Health Care Service Delivery Processes: A Systematic Review and Meta-analysis. *PLoS Med*, 10(1):e1001363.

Manzoni, G.M., Pagnini, F., Corti, S. et al., 2011. Internet-based Behavioral Interventions for Obesity: An Updated Systematic Review. *Clinical Practice Epidemiology Mental Health*, 7:19-28.

Wang, Y.C., McPherson, K., Marsh, T. et al. 2011. Health and Economic Burden of the Projected Obesity Trends in the USA and the UK. *Lancet,* 378(9793): 815-825.

Williams, S.L., and French, D.P. 2011. What Are the Most Effective Intervention Techniques for Changing Physical Activity Self-efficacy and Physical Activity Behavior—and Are They the Same? *Health Education Research*, 26(2): 308–22.

World Health Organization [WHO]. 2013. Obesity and Overweight Fact Sheet. Available at http://www.who.int/mediacentre/factsheets/fs311/en/.

Customers' Retention Requires an Explainability Feature in Machine Learning Systems They Use

Boris Galitsky

Oracle Corp. Redwood Shores CA USA
boris.galitsky@oracle.com

Abstract

We formulate a question of how important explainability feature is for customers of machine learning (ML) systems. We analyze the state of the art and limitations of explainable and unexplainable ML. To quantitatively estimate the volume of customers who request explainability from companies employing ML systems, we analyze customer complaints. We build a natural language (NL) classifier that detects a request to explain in implicit or explicit form, and evaluate it on the set of 800 complaints. As a result of classifier application, we discover that a quarter of customers demand explainability from companies, when something went wrong with a product or service and it has to be communicated properly by the company. We conclude that explainability feature is more important than the recognition accuracy for most customers.

Introduction: Accuracy vs Explainability

Machine learning (ML) has been successfully applied to a wide variety of fields ranging from information retrieval, data mining, and speech recognition, to computer graphics, visualization, and human-computer interaction. However, most users often treat a machine learning model as a black box because of its incomprehensible functions and unclear working mechanism (Liu et al., 2017). Without a clear understanding of how and why a model works, the development of high performance models typically relies on a time-consuming trial-and-error process. As a result, academic and industrial ML scientists are facing challenges that demand more transparent and explainable systems for better understanding and analyzing ML models, especially their inner working mechanisms.

The question of whether accuracy or explainability prevails in an industrial machine learning systems is fairly important. The best classification accuracy is typically achieved by black-box ML models such as Support Vector Machine, neural networks or random forests, or complicated ensembles of all of these. These systems are referred to as black-boxes and their drawbacks are frequently cited since their inner workings are really hard to understand. They do not usually provide a clear explanation of the reasons they made a certain decision or prediction; instead, they just output a probability associated with a prediction. The major problem here is that these methods typically require extensive training sets.

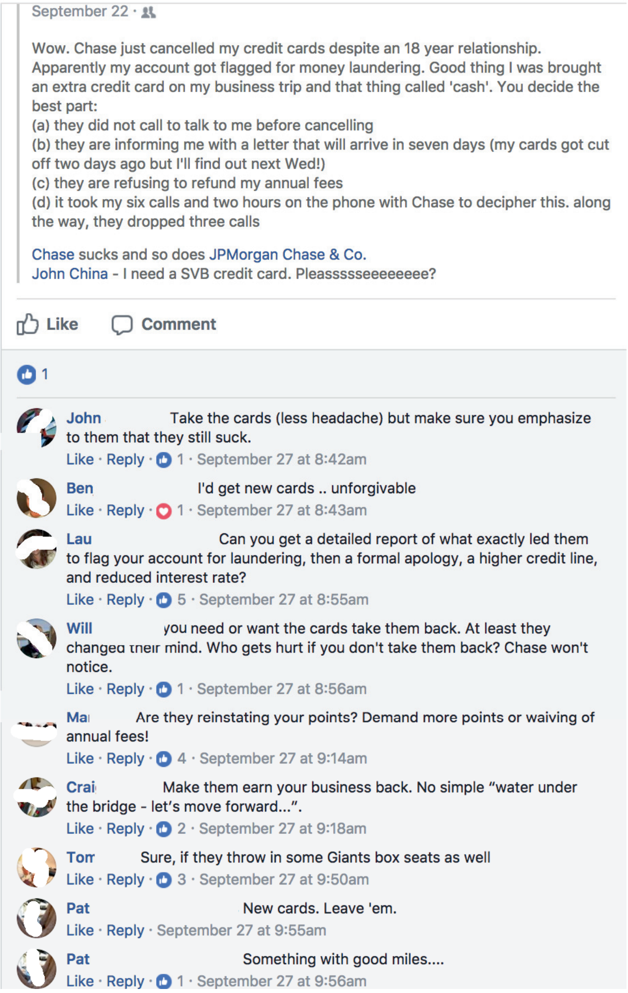
On the other hand, ML methods whose predictions are easy to understand and interpret frequently have limited predictive capacity (inductive inference, linear regression, a single decision tree) or are inflexible and computationally cumbersome, such as explicit graphical models. These methods usually require less data to train from.

Our claim in this study for industrial applications of ML is as follows. Whereas companies need to increase an overall performance for the totality of users, individual users mostly prefer explainability. Users can tolerate wrong decisions made by the companies' ML systems as long as they understand why these decisions were made. Customers understand that any system is prone to errors, and they can be positively or negatively impressed by how a company rectifies these errors. In case an error is made without an explanation, and could not be fixed reasonably well and communicated properly, customers frequently want to leave the business.

We will back up this observation, automatically analyzing customer complaints. To do that, we develop a machinery to automatically classify customer complaints with respect to whether explanation was demanded or not. This is a nontrivial problem since complaint authors do not always explicitly write about their intent to request explanation. We then compare the numbers of customers complaining about problems associated with products and services and estimate the proportion of those complaints, which require explanations.

Use Cases for the ML System Lacking Explainability

Although ML is actively deployed and used in industry, user satisfaction is still not very high in most domains. We will present three use cases where explainability and interpretability of machine learning decisions is lacking and users experience dissatisfaction with certain cases.



September 22 ·

Wow. Chase just cancelled my credit cards despite an 18 year relationship. Apparently my account got flagged for money laundering. Good thing I was brought an extra credit card on my business trip and that thing called 'cash'. You decide the best part:
(a) they did not call to talk to me before cancelling
(b) they are informing me with a letter that will arrive in seven days (my cards got cut off two days ago but I'll find out next Wed!)
(c) they are refusing to refund my annual fees
(d) it took my six calls and two hours on the phone with Chase to decipher this. along the way, they dropped three calls

Chase sucks and so does JPMorgan Chase & Co.
John China - I need a SVB credit card. Pleassssseeeeeeee?

John Take the cards (less headache) but make sure you emphasize to them that they still suck.
Like · Reply · 1 · September 27 at 8:42am

Ben I'd get new cards .. unforgivable
Like · Reply · 1 · September 27 at 8:43am

Lau Can you get a detailed report of what exactly led them to flag your account for laundering, then a formal apology, a higher credit line, and reduced interest rate?
Like · Reply · 5 · September 27 at 8:55am

Will you need or want the cards take them back. At least they changed their mind. Who gets hurt if you don't take them back? Chase won't notice.
Like · Reply · 1 · September 27 at 8:56am

Mai Are they reinstating your points? Demand more points or waiving of annual fees!
Like · Reply · 4 · September 27 at 9:14am

Crai Make them earn your business back. No simple "water under the bridge - let's move forward...".
Like · Reply · 2 · September 27 at 9:18am

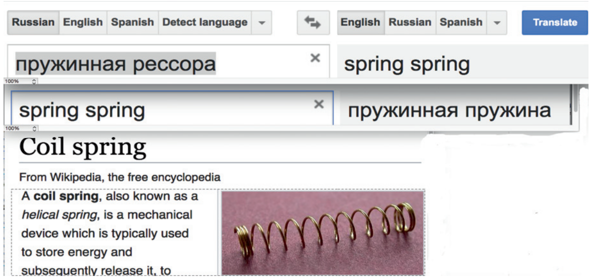
Tom Sure, if they throw in some Giants box seats as well
Like · Reply · 3 · September 27 at 9:50am

Pat New cards. Leave 'em.
Like · Reply · September 27 at 9:55am

Pat Something with good miles....
Like · Reply · 1 · September 27 at 9:56am

Figure 1: A customer is confused and his peers are upset when his credit card is canceled but no explanation is provided.

A customer of financial services are appalled when they travel and their credit cards are canceled without an obvious reason (Fig. 1). The customer explains what had happened in details and his Facebook friends strongly support his case again the bank. Not only the banks made an error in its decision, according to what the friends write, but also it is unable to rectify it and communicate it properly.



пружинная рессора → spring spring

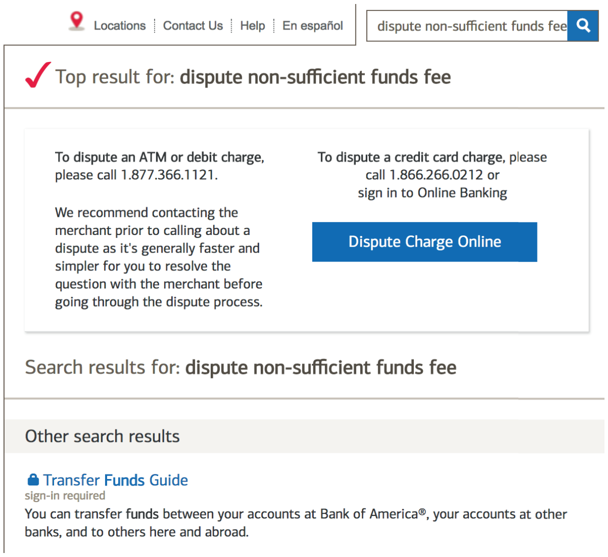
spring spring → пружинная пружина

Coil spring

From Wikipedia, the free encyclopedia

A **coil spring**, also known as a *helical spring*, is a mechanical device which is typically used to store energy and subsequently release it, to

Figure 2: Google translation results for a simple phrase shows the problems in handling context.



✓ Top result for: **dispute non-sufficient funds fee**

To dispute an ATM or debit charge, please call 1.877.366.1121.

We recommend contacting the merchant prior to calling about a dispute as it's generally faster and simpler for you to resolve the question with the merchant before going through the dispute process.

To dispute a credit card charge, please call 1.866.266.0212 or sign in to Online Banking

Dispute Charge Online

Search results for: **dispute non-sufficient funds fee**

Other search results

Transfer Funds Guide
sign-in required
You can transfer funds between your accounts at Bank of America®, your accounts at other banks, and to others here and abroad.

Figure 3: Search engine shows results very far from what a user is asking and do not attempt to explain how they were obtained.

If this bank used a decision making system with explainability, there would be a given cause of its decision. Once it is established that this cause does not hold, the bank is expected to be capable of reverting its decision efficiently and retaining the customer.

An example of a popular machine learning system is shown in Fig. 2. The system translates the term *coil spring* (in Russian) into *spring spring*. This example shows problem in the simplest case of translation where a meaning of two words needs to be combined. A simple meta-reasoning system, a basic grammar checking component or an entity lookup would prevent this translation error under appropriate compartmental ML architecture with explainability. However, a black-box implementation of machine translation breaks even in simple cases like this. Inverse translation is obviously flawed as well (in the middle of Fig. 2).

The bottom shows the fragment of a Wikipedia page for the entity.

Search engine is another application area for ML where relevance score is a major criterion to show certain search results (Fig. 3). Having a highest relevance score does not provide an explanation that the results are indeed relevant. Typical relevance score such as TF*IDF is hardly interpretable; search highlighting features are helpful but the search engine needs to be able to explain *why* it ignored certain keywords like *non-sufficient funds*. A better phrase handling would also help: the system should recognize the whole expression *non-sufficient funds fee* and if it does not occur in search results, explain it.

To investigate how important it is for a customer to have a company's decision explained, to have a decision associated with financial service *interpretable and compatible with common sense*, we need the following. A high number of scenarios of financial service failure have to be accumulated and a proportion of those requiring explanation from the company in one form or another have to be assessed. To do that, we form a dataset of customer complaint scenarios and build an automated assessment framework to detect the cases where explainability is requested.

The Dataset for Tracking Explainability Intent

The purpose of this dataset is to obtain texts where authors do their best to bring their points across by employing all means to show that they (as customers) are right and their opponents (companies) are wrong (Galitsky et al 2009). Complainants are emotionally charged writers who describe problems they encountered with a financial service, lack of clarity and transparency as this problem was communicated with customer support personnel, and how they attempted to solve it. Raw complaints are collected from PlanetFeedback.com for a number of banks submitted over last few years. Four hundred complaints are manually tagged with respect to perceived complaint validity, proper argumentation, detectable misrepresentation, and whether request for explanation concerning the company's decision occurred.

Judging by complaints, most complainants are in genuine distress due to a strong deviation between what they expected from a service, what they received, how this deviation was explained and how the problem was communicated by a customer support. Most complaint authors report incompetence, flawed policies, ignorance, lack of common sense, inability to understand the reason behind the company's decision, indifference to customer needs and misrepresentation from the customer service personnel. The authors are frequently confused, looking for company's explanation, seeking recommendation from other users and advise others on avoiding particular financial service. The focus of a complaint is a proof that the proponent is right and her opponent is wrong, suggested explanation for why the company decides to act in a certain way, a resolution proposal and a desired outcome.

Multiple argumentation patterns are used in complaints. We are interested in argumentation patterns associated with explainability: *I can explain (communicate why I did) it but my opponent (the company) cannot.*

The most frequent is a deviation from what has happened from what was expected, according to common sense. This pattern covers both valid and invalid argumentation.

The second in popularity argumentation patterns cites the difference between what has been promised (advertised, communicated) and what has been received or actually occurred. This pattern also mentions that the opponent does not play by the rules (valid pattern).

A high number of complaints are explicitly saying that bank representatives are lying. Lying includes inconsistencies between the information provided by different bank agents, factual misrepresentation and careless promises (valid pattern).

Another reason complaints arise is due to rudeness of bank agents and customer service personnel. Customers cite rudeness in both cases, when the opponent point is valid or not (and complaint and argumentation validity is tagged accordingly). Even if there is neither financial loss or inconvenience the complainants disagree with everything a given bank does, if they been served rudely (invalid pattern).

Complainants cite their needs as reasons bank should behave in certain ways. A popular argument is that since the government via taxpayers bailed out the banks, they should now favor the customers (invalid pattern).

Complaint authors reveal shady practice of banks during the financial crisis of 2007, such as manipulating an order of transactions to charge a highest possible amount of non-sufficient fund fees. Moreover, banks attempted to communicate this practice as a necessity to process a wide amount of checks. This is the most frequent topic of customer complaints, so one can track a manifold of argumentation patterns applied to this topic.

For most frequent topics of complaints such as insufficient funds fee or unexpected interest rate rise on a credit card, this dataset provides many distinct ways of argumentation that this fee is unfair. Therefore, this dataset allows for systematic exploration of the peculiar topic-independent clusters of argumentation patterns such as a request to explain why certain decision was made. Unlike professional writing in legal and political domains, authentic writing of complaining users have a simple motivational structure, a transparency of their purpose and occurs in a fixed domain and context. Arguments play a critical rule for the well-being of the authors, subject to an unfair charge of a large amount of money or eviction from home. Therefore, the authors attempt to provide as strong argumentation as possible to back up their claims and strengthen their case.

The tag in this dataset used in the current study, request for explanation, is related to the whole text of complaint, not a paragraph. Three annotators worked with this dataset, and inter-annotator agreement exceeds 80%. The set of tagged customer complaints about financial services is available at https://github.com/bgalitsky/relevance-based-on-parse-trees/blob/master/examples/opinionsFinanceTags.xls .

Automated Detection of a Request to Explain

Obviously, just relying on keywords, using keyword rules is insufficient to detect implicit request to explain. Hence an ML approach is required with the training dataset with text including a request to explain and not including one. Not just syntax level but discourse-level features are required when a request to explain is not explicitly mentioned. We select the Rhetoric Structure Theory (Rhetoric Structure Theory (RST, Mann and Thompson 1988) as a means to represent discourse features associated with affective argumentation.

We use an example of a request to recommend & explain to demonstrate a linguistic structure for explainability (Fig. 4). This text (from a collection of odd questions to Yahoo! Answers) is a question that expects not just a brief answer “do this and do that” but instead a full recommendation with explanation:

I just had a baby and it looks more like the husband I had my baby with. However it does not look like me at all and I am scared that he was cheating on me with another lady and I had her kid. This child is the best thing that has ever happened to me and I cannot imagine giving my baby to the real mom.

The chain of rhetoric relations *RST-elaboration* (default), *RST–sequence* and *RST-contrast* indicate that a question is not just enumeration of topics and constraints for an expected answer (that can be done by *RST-elaboration* only). Instead, this chain indicates that a conflict (an expression that something occurs in contrast to something else) is outlined in a question, so an answer should necessarily include an explanation.

We combine parse trees for sentences with pragmatic and discourse-level relationships between words and parts of the sentence in one graph, called parse thicket (Galitsky 2012). We complement the edges for syntactic relations obtained and visualized with the Stanford NLP system (Manning et al., 2014). For coreferences, (Recasens et al., 2013 and Lee at al 2013) was used. The arcs for pragmatic and discourse relations, such as anaphora, same entity, subentity, rhetorical relation and communicative actions are manually drawn in red. Labels embedded into arcs denote the syntactic relations.

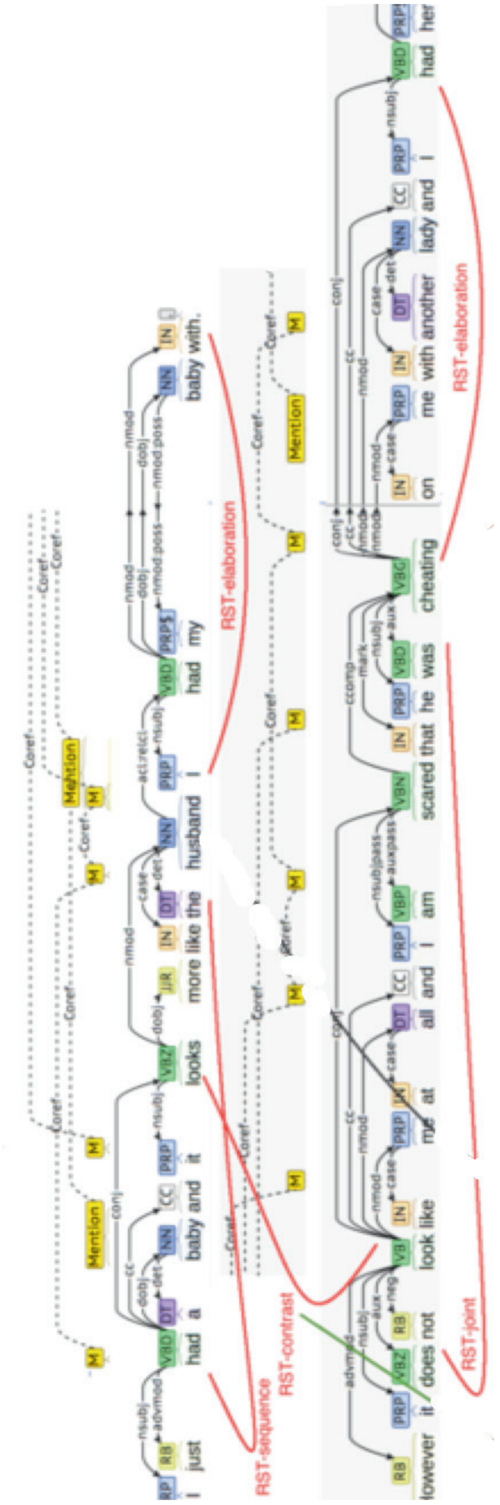

RST-elaboration
RST-sequence
RST-contrast
RST-joint
RST-elaboration
Mention
Mention
Mention
Coref
I just had a baby and it looks more like the husband I had my baby with.
However it does not look like me at all and I am scared that he was cheating on me with another lady and I had her

Figure 4: Linguistic representation for text which contains a request to explain.

Lemmas are written below the boxes for the nodes, and parts-of-speech are written inside the boxes.

This graph includes much richer information than just a combination of parse trees for individual sentences would. Navigation through this graph along the edges for syntactic relations as well as arcs for discourse relations allows to transform a given parse thicket into semantically equivalent forms to cover a broader spectrum of possibilities to express a request to explain.

To form a complete formal representation of a paragraph, we attempt to express as many links as possible: each of the discourse arcs produces a pair of thicket phrases that can be a potential match with an expression for explainability request. Further details on using nearest neighbor learning via maximal common sub- parse thicket are available in (Galitsky 2012).

Evaluation of Recognition Accuracy and Assessment of the Proportion of Request to Explain

Once we developed our algorithm for explanation request detection, we want to train it, test it and verify how consistent its results are across the domains. We also test how recognition accuracy varies for cases of different complexity.

Table 1: Cases of explanation requests and detection accuracies for model development and evaluation

Evidence	*#*	*Criteria*	*P*	*R*	*F1*
Imperative expression with communicative action *explain*	44	Keywords: explain, clarify, make clear, why did they act-VP, why was it	92	94	93.0
Double, triple+ implicit mention	67	Multiple rhetoric relation of *contrast, attribution, sequence, cause*	86	83	84.5
Single implicit mention	115	A pair of rhetoric relation chains for *contrast* and *cause*	76	80	77.9

Detection accuracy for explanation request for different types of evidence is shown in Table 1. We consider simpler cases where the detection occurs based on phrases, in the top row. Typical expressions here have an imperative form such as *please explain/clarify/motivate/comment.* Also, there are templates here such as you *did this but I expected that … you told me this but I received that.*

The middle row contains the data on higher evidence implicit explanation request case, where multiple fragments of DTs indicate the class. Finally, in the bottom row, we present the case of the lower confidence for a single occurrence of a DT associated with an explanation request. The second column shows the counts of complaints per case. The third column gives examples of expressions (which include keywords and phrase types) and rhetorical relations which serve as criteria for implicit explanation request. Fourth, fifth and sixth columns presents the detection rates where the complaints for a given case is mixed with a hundred of complaints without explanation request.

Recognition accuracies, bank-specific topics of complaints and an overall proportion of the complaints with explanation request are shown in Table 2. We used 200 complaints for each bank to assess the recognition accuracies for explanation request (ER). One can observe that 82±3% is a reasonable estimate for recognition accuracy for explanation request. The last column shows that taking into account <20% error rate in explanation request recognition, 25±4% is an adequate estimate of complaints requiring explainability in implicit or explicit form, given the set of 800 complaints.

Table 2: Discovering explanation request rates for four banks

Source	*#*	*Main topics of complaints*	*P*	*R*	*F1*	*ER rate*
Bank of America	200	NSF, credit card interest rate raise	82	84	83.0	28.5
Chase Bank	200	NSF, foreclosure , unexpected card cancellation	80	82	81.0	25.8
Citibank	200	Foreclosure, mortgage application, refinancing,	79	83	81.0	23.8
American Express	200	Card application, NSF, late payment	83	82	82.5	27.0

Finally, we ran our explanation request detection engine against the set of 10000 complaints scraped from PlanetFeedback.com and observed that 27% of complainants explicitly or implicitly require explainability from companies for their decisions. There is a single complaint per author. Our observation is that since almost a quarter of customers strongly demand and rely on explainability of the companies' decisions, these customers are strongly affected by the lack of explainability and may want to switch to another service. Hence the companies need to employ ML algorithms with explainability feature. A very small number of customers complained about errors in decisions irrespectively of how these errors were communicated (a manual analysis). Hence we conjecture that customers are affected by a lack of explainability in a much higher degree than by an error rate (such as extra 10%, based on anecdotal evidence) of a company's decision-making system.

This explainability feature is more important than the recognition accuracy for the customers, who understand that all businesses make errors. Typically, when a company makes a wrong decision via ML but then rectifies it efficiently, a complaint does not arise. The most important means for customer retention is then properly communicating with them both correct and possibly erroneous customer decisions (not quantitatively evaluated in this study).

Related Work

To tackle the challenges associated with the lack of explainability of most popular modern ML algorithms, there are some initial efforts on interactive model analysis. These efforts have shown that interactive visualization plays a critical role in understanding and analyzing a variety of machine learning models. Recently, DARPA I2O released Explainable Artificial Intelligence proposal to encourage research on this topic. The main goal of XAI is to create a suite of machine learning techniques that produce explainable models to enable users to understand, trust, and manage the emerging generation of AI systems.

There have been attempts to augment the learning models intrinsically lacking explainability with this feature. ML models can be trained to automatically map documents into abstract concepts such as semantic category, writing style, or sentiment, allowing to categorize a large corpus. Besides predicting the text's category, it is essential to understand how the categorization process arrived to a certain value. (Arras et al., 2017) demonstrate that such understanding can be achieved by tracing the classification decision back to individual words using layer-wise relevance propagation, a recently developed technique for explaining predictions of complex non-linear classifiers. The authors trained two word-based ML models, a CNN and a bag-of-words SVM classifier, on a topic categorization task and applied the layer-wise relevance propagation method to decompose the predictions of these models onto words. Resulting scores indicate how much individual words contribute to the overall classification decision. This enables one to distill relevant information from text documents without an explicit semantic information extraction step. The authors further used the word pair-wise relevance scores for generating novel vector-based document representations which capture semantic information. Based on these document vectors, a measure of model explanatory power was introduced and showed that, although the SVM and CNN models perform similarly in terms of classification accuracy, the latter exhibits a higher level of explainability which makes it more comprehensible for humans and potentially more useful for other applications.

Although ML models are widely used in many applications due to high accuracy, they fail to explain their decisions and actions to users. Without a clear understanding, it may be hard for users to leverage their knowledge by their learning process and achieve a better prediction accuracy. As a result, it is desirable to develop more explainable machine learning models, which have the ability to explain their rationale and convey an understanding of how they behave in the learning process. The key challenge here is to design an explanation mechanism that is tightly integrated into the ML model. Accordingly, one interesting future work is to discover which parts in an ML model structure explains its different functions and plays a major role in the performance improvement or decline at each iteration. One possibility is to better back up both the model and the decisions made. In particular (Lake et al., 2015) proposed a probabilistic program induction algorithm, having developed a stochastic program to represent concepts, which are formed compositionally from parts and spatial relations. (Lake et al., 2015) showed that their algorithm achieved human-level performance on a one-shot classification task, However, for the tasks that have abundant training data, such as object and speech recognition, CNN approaches still outperform (Lake et al., 2015) algorithm. There is still a long path to proceed towards more explainable deep learning decisions.

Following the recent progress in deep learning, ML scientists are recognizing the importance of understanding and interpreting what goes on inside these black box models. RNN have recently improved speech recognition and translation, and these powerful models would be very useful in other applications involving sequential data. However, adoption has been slow in domains such as jurists, finance, legal and health, where current specialists are reluctant to let an explanation-less engine make crucial decisions. (Krakovna et al., 2016) suggests to make the inner workings of RNNs more interpretable so that more applications can benefit from their power.

CNNs have achieved breakthrough performance in many pattern recognition tasks such as image classification. However, the development of high-quality deep models typically relies on a substantial amount of trial-and-error, as there is still no clear understanding of when and why a deep model works. (Liu et al 2017) presents a visual analytics approach for better understanding, diagnosing, and refining deep CNNs. The authors simulated CNN as a directed acyclic graph. Based on this formulation, a hybrid visualization is developed to visualize the multiple facets of each neuron and the interactions between them. The authors also introduced a hierarchical rectangle-packing algorithm and a matrix re-shuffling method to show the derived features of a neuron cluster. They also proposed a bi - clustering-based edge merging algorithm to minimize visual distortion caused by a large number of connections between neurons.

Conclusions

We demonstrated that customers are strongly dissatisfied when decisions strongly affecting them are made by ML systems lacking explainability and interpretability features. Popularity of deep learning approaches, which make these features harder to implement, further increase customer dissatisfaction and negatively affect their user retention. Whereas deep learning and big data approaches decrease the development costs for companies, especially when sufficient data is available, customer satisfaction drops.

We developed the NL detection system for the case of explainability request and detected 27% of complaints require explainability from companies, from the set of 10000 complaints. Hence a quarter of customers strongly demand and rely on explainability of the companies' decisions (Galitsky 2017). The conclusion is that these customers are strongly affected by a lack of explainability of ML system and would stop being customers if a competitive business offers explanation-enabled service. Hence we conjecture that the companies need to employ ML algorithms with explainability feature.

References

Turner, Ryan. 2016. A Model Explanation System: Latest Updates and Extensions, CoRR abs/1606.09517.

Arras, Leila, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, Wojciech Samek. 2017. What is relevant in a text document?: An interpretable machine learning approach. PloS one 10.1371/journal.pone.0181142.

DARPA. 2016. Explainable artificial intelligence (XAI). http://www.darpa.mil/program/explainable-artificial-intelligence (last downloaded November 2017).

Galitsky, B. and Josep Lluis de la Rosa. 2011. Concept-based learning of human behavior for customer relationship management. Information Sciences. Volume 181, Issue 10, 15 May 2011, pp 2016-2035.

Galitsky, B., MP González, CI Chesñevar 2009. A novel approach for classifying customer complaints through graphs similarities in argumentative dialogue. Decision Support Systems, 46-3, 717-729.

Galitsky, B. 2016. Using extended tree kernels to recognize metalanguage in text. Uncertainty Modeling, in Kreinovich V., editor. Springer.

Galitsky, B. 2012. Machine learning of syntactic parse trees for search and classification of text. Engineering Application of AI . Volume 26, Issue 3, Pages 1072-1091.

Galitsky 2017. Natural Language Classifier that Detects a Request to Explain and Generates a Natural Language Response. Oracle Patent Application ORA180478.

Krakovna, Viktoriya and Finale Doshi-Velez. 2016. Increasing the Interpretability of Recurrent Neural Networks Using Hidden Markov Models. CoRR. abs/1606.05320.

Liu, Mengchen, Jiaxin Shi, Zhen Li, Chongxuan Li, Jun Zhu, and Shixia Liu. 2017. Towards Better Analysis of Deep Convolutional Neural Networks. IEEE Transactions on Visualization and Computer Graphics 23, 1 (January 2017), 91-100. DOI: https://doi.org/10.1109/TVCG.2016.2598831.

Recasens, Marta, Marie-Catherine de Marneffe, and Christopher Potts. 2013. The Life and Death of Discourse Entities: Identifying Singleton Mentions. In Proceedings of NAACL.

Heeyoung Lee, Angel Chang, Yves Peirsman, Nathanael Chambers, Mihai Surdeanu and Dan Jurafsky. 2013. Deterministic coreference resolution based on entity-centric, precision-ranked rules. Computational Linguistics 39(4).

Mann, William and Sandra Thompson. 1988. Rhetorical structure theory: Towards a functional theory of text organization. Text-Interdisciplinary Journal for the Study of Discourse, 8(3):243–281.

Lake, B. M. , Salakhutdinov, R. and J. B. Tenenbaum, 2015. Human-level concept learning through probabilistic program induction, Science 350 (6266) 1332–1338.

Retrieval System for Data Utilization Knowledge Integrating Stakeholders' Interests

Teruaki Hayashi, Yukio Ohsawa
The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, Tokyo, Japan

Abstract

In the worldwide trend of big data and AI, the use of secondary data is an essential social demand in terms of cross-discipline data exchange and utilization. However, the intentions or the contexts of data collected by others tend to be unclear, which may cause difficulties in grasping facts for decision making. To avoid this contextual gap between users and collectors, we propose the retrieval system of data reflecting the collectors' contexts as well as users' requirements for making a beneficial decision. Our motivation is to support users to retrieve data without extensive knowledge of them and to improve the transparency of the retrieval process. To achieve these aims, we conducted several workshops to collect integrated knowledge of data utilization and implemented an interface for users to retrieve information about data related to their interests from free text queries.

Introduction

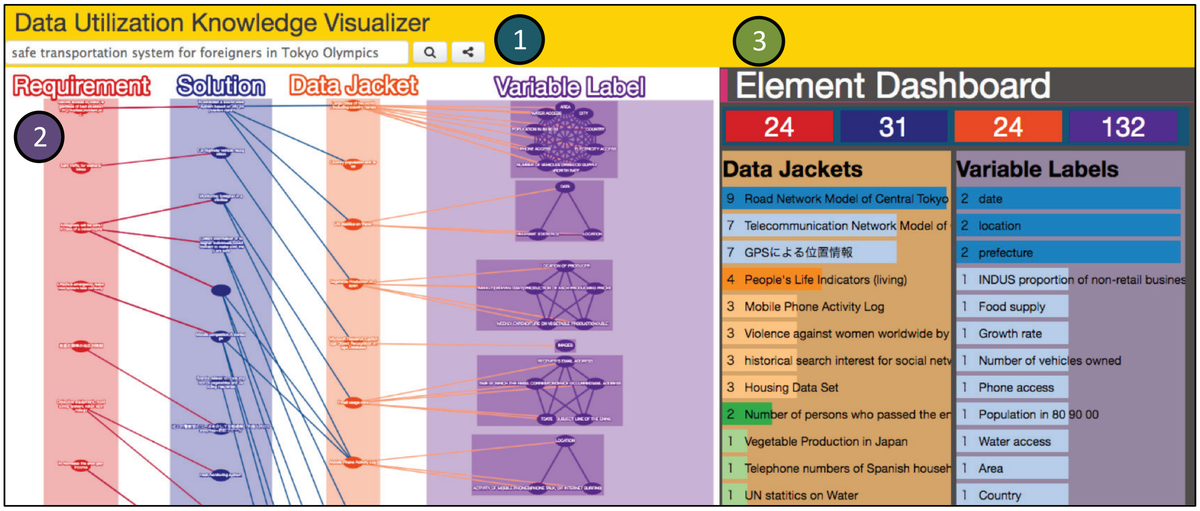
It is difficult for users to accurately obtain data corresponding to their intentions because they struggle to express their objects of interest using the terms in the relevant data. This problem occurs owing to a lack of knowledge of the data and a lack of knowledge about how to use data. To tackle such problems, we propose herein an interactive visualization and retrieval system by structuring knowledge relating to data utilization. In this study, knowledge of data utilization consists of three types of entities: Requirements, Solutions, and Data Jackets. Requirements are the problems stated by users, Solutions are the proposals for analysis with combinations of data that satisfy the Requirements, and Data Jackets represent information relating to data. Data Jacket (DJ) is a technique for sharing "a summary of data" as metadata without sharing the data itself (Ohsawa et al. 2013).

System Architecture

The three layers to describe the knowledge of data utilization are based on the three roles of the players in the market of data: the data owner, the user, and the analyst. Considering that the data owners offer DJs, the data users state Requirements, and the data analysts propose Solutions, two minimum units of knowledge of data utilization can be described as follows with binary predicate logic: $\mathbf{combine}(solution, DJ)$ and $\mathbf{satisfy}(solution, requirement)$. We can describe complex knowledge of data utilization structurally by combining these models, which consist of a 3-partite graph. For example, the retrieval system returns a set of DJs constituting Solutions satisfying the corresponding Requirements, from a query to the Requirement database. Also, by adding the information about variables (Variable Labels: VLs) to the 3-partite graph, it is possible to learn the combinations of variables constituting data. In general, a VL is a concrete entity but a Requirement is an abstract entity. Both are different types of knowledge elements and there is no linkage to bridge Requirements and VLs. Introducing knowledge of data utilization, the system can retrieve VLs by way of Solutions and DJs. Furthermore, the retrieval process is clear through visualization, which enhances the persuasiveness of the search results (Sinha and Swearingen 2002). To create a knowledge base of data utilization, we conducted an Innovators Marketplace on Data Jackets (IMDJ). The IMDJ is a workshop for discussing data utilization using DJs. Data owners provide their datasets as DJs, and participants of the IMDJ (data owners, users, and analysts) create Solutions to solve users' problems stated as Requirements. We stored 190 DJs, 276 Solutions, and 222 Requirements created in 16 IMDJ workshops as the knowledge base of data utilization.

Fig.1 is the interface showing the 3-partite graph of knowledge of data utilization with VLs. The figure shows the results of retrieval from the free text query "safe transportation system for foreigners in Tokyo Olympics" using a retrieval algorithm. Users send queries using the text area ① and the result is visualized in ②. The area ③ contains detailed information related to the result (the numbers of retrieved knowledge elements). The visualization of the 3-partite graph is represented based on a directed graph $G_W = (V_W, E_W)$ (W: a set of knowledge elements, V_W: a set of nodes, E_W: a set of edges). The co-occurrence graph of VLs is represented based on a weighted undirected graph $G_S = (V_S, E_S)$ ($V_S = \{vl_i \in S\}$, $E_S = \{(vl_i, vl_j)_{dj_k} | vl_i, vl_j \in S, vl_i \neq vl_j\}$).

When users entered free text (T_x) as a query, the system searched for the database of DJs and returned a set of DJs ($DJ_{dj(T_x)} = \bigcup_{i=1}^{T} DJ_{dj(t_i)}$) obtained by string matching a word (t_i) with the descriptions of the DJs. Likewise, when searching for DJs in the Solutions, the system searched for



24	31	24	132

Data Jackets		Variable Labels	
9	Road Network Model of Central Tokyo	2	date
7	Telecommunication Network Model of	2	location
7	GPSによる位置情報	2	prefecture
4	People's Life Indicators (living)	1	INDUS proportion of non-retail busines
3	Mobile Phone Activity Log	1	Food supply
3	Violence against women worldwide by	1	Growth rate
3	historical search interest for social netv	1	Number of vehicles owned
3	Housing Data Set	1	Phone access
2	Number of persons who passed the er	1	Population in 80 90 00
1	Vegetable Production in Japan	1	Water access
1	Telephone numbers of Spanish househ	1	Area
1	UN statitics on Water	1	Country

Figure 1: The interface of our retrieval system.

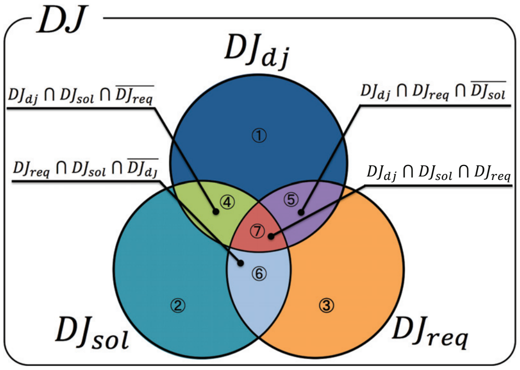

DJ
DJ_{dj}
$DJ_{dj} \cap DJ_{sol} \cap \overline{DJ_{req}}$
$DJ_{dj} \cap DJ_{req} \cap \overline{DJ_{sol}}$
$DJ_{req} \cap DJ_{sol} \cap \overline{DJ_{dj}}$
$DJ_{dj} \cap DJ_{sol} \cap DJ_{req}$
①
④
⑤
⑦
⑥
②
③
DJ_{sol}
DJ_{req}

Figure 2: The types of DJ sets.

the database of Solutions by checking for strings matching a word (t_i) with the descriptions of the Solutions, and returned a set of DJs ($DJ_{sol(T_x)}$) connected with the Solutions. Similarly, when retrieving DJs from Requirements, the system searched for the database of Requirements by matching the string represented by a word (t_i) with the descriptions of the Requirements and returned a set of Solutions connected to the Requirements. The system then searched for DJs connected to the set of Solutions and returned a set of DJs ($DJ_{req(T_x)}$). Finally, the system returned a set of DJs ($DJ_{dj(T_x)} \cup DJ_{sol(T_x)} \cup DJ_{req(T_x)}$).

Results and Conclusion

To evaluate the system's ability to retrieve information potentially related to users' interests, we collected 4,326 search queries from the users' searching behaviors on the portal site of DJs. We compared the number of DJs in the DJ group and the Req-Sol group retrieved by queries. The DJ group are the DJs found by the string matching of words in the DJ database (areas ①, ④, ⑤, and ⑦ in Fig.2), and the Req-Sol group are the DJs found the string matching of words only in the Solution or the Requirement database (areas ②, ③, and ⑥ in Fig. 2). The result shows that a larger number of DJs can be retrieved by way of the Solution and the Requirement databases than by searching only the DJ database (Fig.3). The result shows that the structured knowledge of data utilization may support the significant discovery of data related to users' interests, even if users cannot adequately express their interests by employing terms found in the data, or do not have sufficient knowledge of the data.

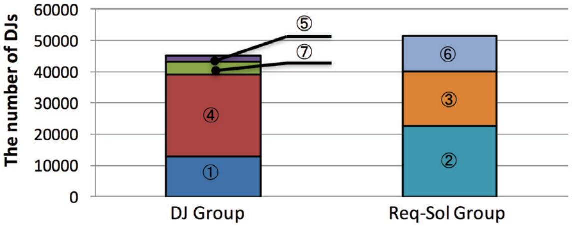


Figure 3: The number of retrieved DJs (the numbers correspond to the numbers in Fig. 2.)

Acknowledgments

This study was partially supported by JST-CREST Grant Number JPMJCR1304, and JSPS KAKENHI Grant Numbers JP16H01836 and JP16K12428.

References

Ohsawa, Y.; Kido, H.; Hayashi, T.; and Liu, C. 2013. Data jackets for synthesizing values in the market of data. *Procedia Computer Science* 22:709–716.

Sinha, R., and Swearingen, K. 2002. The role of transparency in recommender systems. In *Extended Abstracts on Human Factors in Computing Systems*, 830–831. ACM.

Policy Decision Support System in Aging Society based on Probabilistic Latent Spatial Semantic Structure Modeling

Ayae Ide
School of computing
Tokyo Institute of Technology
Tokyo, Japan
ide.ayae@aist.go.jp

Yoichi Motomura
Artificial Intelligence Research Center
National Institute of
Advanced Industrial Science and Technology
Tokyo, Japan
y.motomura@aist.go.jp

Takao Terano
School of computing
Tokyo Institute of Technology
Tokyo, Japan
terano@dis.titech.ac.jp

Abstract

This paper analyzes a questionnaire survey data on elderly people in order to investigate regional characteristics of their living activities. For the purpose, we use Probabilistic Latent Spatial Semantic (PLSS) Modeling, which is integrated the two methods: probabilistic latent semantic analysis (pLSA) and Bayesian network (BN). First, we aggregate each individual's survey record by postal code; Second, we find characteristics of the region by pLSA; Third, we use BN to clarify factors of this regional disparity. From the study, we are able to identify critical information to support decisions for a manager in a local government: i) There is regional disparity in terms of social network; and ii) The regional disparity of social network will improve by neighborhood facilities. Such information will be of use for designing the super-aged society in the near future. We propose policy decision support system in aging society based on PLSS Modeling.

Introduction

As the aging will rapidly progress in worldwide, especially in Japan, which is the top of such a super aged society, the policy decision making in Japan is urgent study topic; for example, here will be the medical shortage caused by super aged society, and the increase in social security expenses. However, policy problems were thought to be difficult to deal with in existing social science. Policy problems are difficult to solve because of its complexity. As this complexity, there are four properties; 1)Comprehensiveness; policy problems involve various problems, 2)Reciprocity; policy problems conflict with other problems, economic development and environmental protection, 3)Subjectivity; Framing of policy problems are different depending on position and viewpoint , 4)Dynamics; Policy problems are changing everyday (Akiyoshi et al. 2015). For that reason, we need to construct model to understand structure and context of policy problems, and this model have to respond flexibly to changes of policy problems and reflect knowledge from multiple viewpoints and experts. That is, it is the model which is easy to understand the relationship between variables and reflect domain knowledge and new data. This research aims to policy decision support system for local government in aging society based on modeling which clarify regional disparities.
To realize this model, we applying computational social science techniques to a large scale survey study conducted by Japan Gerontological Evaluation Study (JAGES). JAGES are conducting large-scale questionnaire survey targeting more than 100,000 elderly people to uncover the current geographical status of living activities of elderly people. With this questionnaire, it is possible to acquire data on elderly people from a multifaceted viewpoint such as body, psychology, society. One of main objects in this project is to clarify regional health disparity and regional characteristics in order to support policy making of local governments.

Currently, data analysis using spatial data is increasingly important in the context of policy decision making. Spatial data are used in various policy fields, and in the field of medical policy, it is applied to problems such as factor analysis of mortality rate and correction of regional disparities. In the mid-19th century, J. Snow created a map to find out the spatial ubiquity of the distribution of cholera patients in London (Snow, John 2015). This is a method of space clustering and has been developed in a field called Spatial epidemiology. Time, person, and place are 3 main epidemiologic variables (Pfeiffer, Dirk, et al. 2008), in particular, place is the most important variables in the context of policy decision making. In the scene of policy, municipalities have intervention at regional level. In this research, we use the postal code attached to the questionnaire data as primary key.

Based on this background, this paper analyzes the questionnaire data with recent plural machine learning techniques and simulates policy effect at regional level. First, we apply a clustering technique to extract postal code groups. Second, we find characteristics of the region by and question item groups via probabilistic latent semantic analysis (pLSA). Third, we apply bayesian network (BN) to the data to understand relations among many variables. Based on the obtained model, we carry out causal reasoning in regional health disparity.

The rest of the paper is organized as follows: In Section 2, we describe the data and method in order; in Section3, we give investigation results of each method; Section 4 gives discussion of the study, and finally, Section 5, we give some concluding remarks.

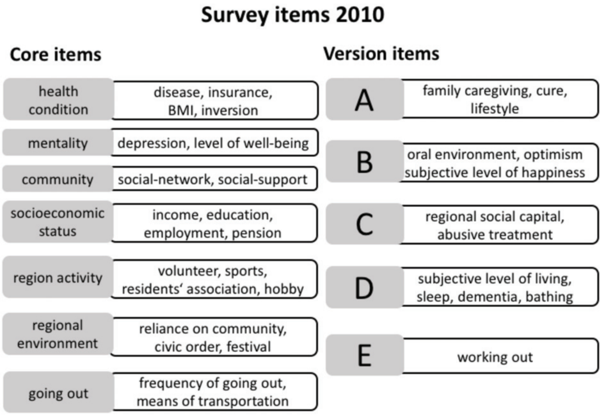

Survey items 2010
Core items
Version items
health condition
disease, insurance, BMI, inversion
mentality
depression, level of well-being
community
social-network, social-support
socioeconomic status
income, education, employment, pension
region activity
volunteer, sports, residents' association, hobby
regional environment
reliance on community, civic order, festival
going out
frequency of going out, means of transportation
A
family caregiving, cure, lifestyle
B
oral environment, optimism subjective level of happiness
C
regional social capital, abusive treatment
D
subjective level of living, sleep, dementia, bathing
E
working out

Figure 1: JAGES question items

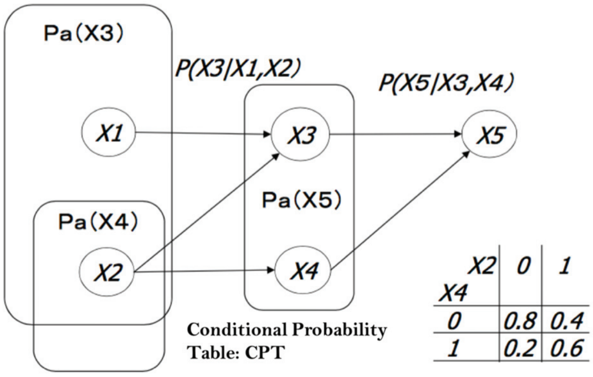


Figure 2: Bayesian network

Data and Methods

Data

The target data of this analysis is JAGES 2010-2013 cohort data which is 2010 cross-sectional data combined with certification of long-term care need in 2013. This is tracking data of the respondents in 24 municipalities targeted for the 2010 survey. There are some variables in certification of long-term care data, for example, dead, dementia and the level of care needed. JAGES 2010-2013 cohort data has 74264 records and 53801 records have postal code.

The question items in the questionnaire consist of core items and version items. Core items are items common to all respondents and five types of version items of the A to E version are equally attached to the core items. The outline of questionnaire items is shown in Fig.1.

Methods

pLSA(Hofmann, Thomas 1999) has been proposed as a method of document classification and is one method of text clustering. In this method, we assume that word w in document d is generated via latent variable z. In the likelihood maximization by the EM algorithm, the latent variable $z \in Z = \{z_1, ..., z_k\}$ is attached to the co-occurring data. In this study, we assume that the postal code w_i responds to the question answer d_i via the latent variable z_k, and extracted postal code groups and question item groups with similar responses in the questionnaire data. Co-occurrence frequency is $n(i, j)$. The joint probability is expressed by the following equation.

$$P(w_i, d_i) = \sum_k P(w_i|z_k)P(d_i|z_k)P(z_k) \tag{1}$$

After that, $P(w|z), P(d|z), P(z)$ are calculated by EM algorithm which maximizes the following log likelihood function.

$$L = \sum_i \sum_j n(i, j) \log P(w_i, d_i) \tag{2}$$

pLSA can maximize information content by EM algorithm so that dimension reduction, which has less loss of information content, can realize in the resulting segment. Analysis that extracted latent classes using pLSA is effective for big data but it does not express explicitly what the extracted latent class represents, so it is difficult to understand the meaning of that latent class intuitively. It is a big problem when manually analyzing after extracting latent classes. Therefore, we consider a probabilistic latent semantic modeling that can model latent classes extracted by pLSA, and furthermore, relationships between latent classes by Bayesian network. By modeling relationships with the explanatory variable, there is an advantage that the latent class which was intuitively difficult to understand can be characterized by the related explanatory variable.

The Bayesian network (Pearl, Judea 1985) is one of graphical model that enables prediction of events and reasonable decision making.The model created can be represented by network graph. The product of simultaneous probabilities among the variables shows the simultaneous distribution of the model. In addition, by using the probabilistic reasoning algorithm, posterior probability calculation, sensitivity analysis can be executed(Motomura 2009).

A Bayesian network is a model in which a qualitative dependency relationship among multiple random variables is represented by a graph structure and a quantitative relationship between individual variables is represented by a preceding conditional probability as Figure2. This is a probabilistic model defined by random variables and the conditional dependency between random variables and its conditional probability. A variable is a node, a dependency relation between variables is represented by an oriented link extending in the direction of the variable resulting from the cause, a node that comes before the link is called a child node, and a node under the link is called a parent node. For example, the dependence relation between random variable X_i and X_j is represented by directed rink X_iX_j. X_i is parent node, and X_j is child node. When we assume that the set of parent nodes $\pi(X_j) = \{X_1, ..., X_i\}$ with child node X_j. The dependency relation between X_j and $\pi(X_j)$ is quantitatively expressed by the following conditional probability.

$$P(X_j|\pi(X_j)) \tag{3}$$

Furthermore, for each of the n random variables $X_1, ..., X_n$, in the same way as a child node, the simultaneous probability distribution of all the random variables is as below.

$$P(X_1, ..., X_n) = \prod_j P(X_j|\pi(X_j)) \tag{4}$$

The Bayesian network is a modeling based on discontinuous probability distribution in which X - Y space is discretized according to the conditional probability table and individual probability values are assigned. However, we apply bayesian network to big data, the number of states of the discrete random variable becomes enormous. For that reason, the size of the conditional probability table becomes huge and frequency distribution becomes sparse, so that model construction becomes difficult. To solve this problem, it is necessary to cluster the state to an appropriate granularity beforehand(Ishigaki et al. 2010). Therefore, clustering by pLSA as prior processing can prevent from frequency distribution becoming sparse(Murayama et al. 2015; Hirokawa et al. 2015). In other words, a structure model corresponding to big data is constructed by classifying the elderly or the region into latent segments with pLSA and constructing a Bayesian network that estimates the probability of belonging to the latent segment from various variables. In this research, we added aggregation by postal code as data processing. We called this method Probabilistic Latent Spatial Semantic (PLSS) Modeling.

Results

pLSA

We extracted latent segments from JAGES dataset aggregated by postal code by using pLSA. The total number of postal codes is 2133.The number of latent class was determined based on AIC(Akaike 1987). There is an initial value dependence because EM algorithm is used for the likelihood calculation of pLSA. Thus, by changing the initial value 5 times, the latent class was increased from 4 to 45 and the minimum value at each cluster number was compared. The figure 3 indicates AIC scores. As a result, since AIC has the minimum value when K = 25, the number of clusters 25 was selected. PLSA is a method for maximum likelihood estimation of topic model so that all variables belong to all clusters and the degree of affiliation is given by probability. We assigned postal codes and question answers to clusters with the highest affiliation probability.In figure 4, this is scatter plot of postal codes in which the vertical line shows $P(w_i|z_k)$ and the horizontal one z_k. As can be seen from the figure, postal codes are distributed mainly in Z003 and Z009 and Z025. Moreover, most of affiliation probabilities is less than 0.5 and it's meaning variables belong to more than one cluster at the same time. Figure 5 and 6 shows affiliation probabilities of all postal codes and question answers in each cluster. All clusters have different distribution of affiliation probability.

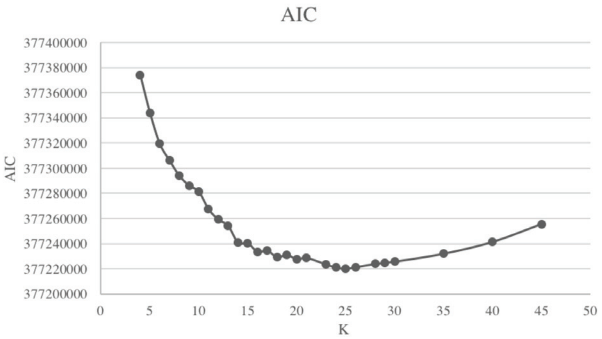


Figure 3: AIC scores

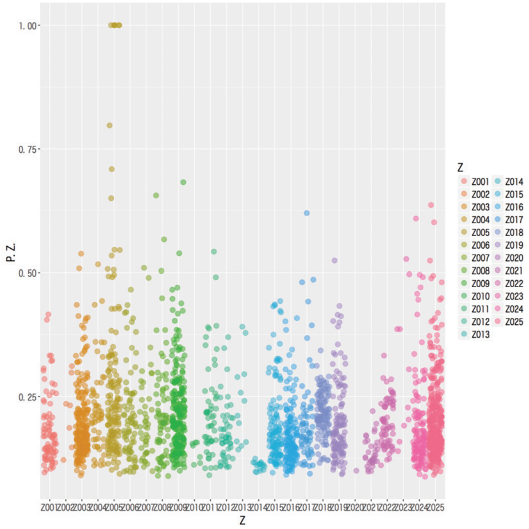


Figure 4: Scatterplot of postal codes with highest affiliation probability

We take these three main clusters for instance. Figure 7 shows top 10 question answers with the highest affiliation probabilities. The columns of question answers are color-coded, in particular, answers about hobby are green, answers about near facilities are blue, and answers about isolation are red. Similar answers are listed for each cluster, and the potential trends of clusters can be interpreted as follows. Z003 and Z009 are the areas where hobby activities are active and no isolation, on the other hand, Z025 is the area where people lacked social and community ties.

We consider Nagoya city which has the largest number of respondents among subject municipalities. First of all, in each 16 wards constituting Nagoya City, the number of postal codes belonging to each 25 clusters was counted. Then, considering the cluster with the highest postal code distribution as the cluster representing the administrative district, a choropleth map is created and shown in figure 8. As shown in figure 8, the representative clusters are equal in the neighboring area. Hence, it was confirmed that the ten-

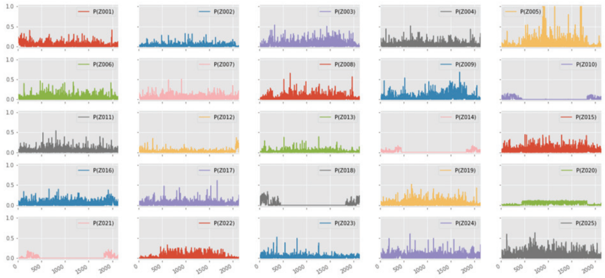


Figure 5: Affiliation probability of postal codes

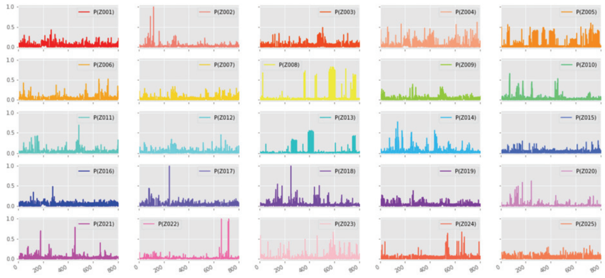


Figure 6: Affiliation probability of question answers

Z003				Z009				Z025			
P(w/z)	Postal code	P(d/z)	question answers	P(w/z)	Postal code	P(d/z)	question answers	P(w/z)	Postal code	P(d/z)	question answers
0.538	4470001	0.487	Hobby (Painting)	0.683	4670035	0.393	Facilities within 1 km from house (Slope and Steps)	0.636	4540868	0.300	Friends or acquaintances frequently meeting (None)
0.508	4640054	0.357	Hobby (Golf)	0.539	4520838	0.324	Hobby(PC)	0.601	0711441	0.285	Hobby (Pachinko)
0.435	4620869	0.341	Receiving pension (Private pension)	0.470	0710251	0.316	Facilities within 1 km from house (Park and Promenade)	0.524	4530846	0.271	Frequency of meeting with friends (None)
0.412	2770083	0.340	Frequency of participation in sports groups(Weekly, Monthly)	0.466	4640836	0.315	Facilities within 1 km from house (Attractive scenery and buildings)	0.494	4530861	0.227	tobacco (smoking cessation within 4 years)
0.410	2770853	0.339	Hobby (Instrument Performance)	0.447	4570864	0.292	Years of school education (over 13 years)	0.489	4540824	0.223	Frequency of eating vegetables and fruits (2–3 per week)
0.408	4670045	0.337	Hobby (tea ceremony・flower arrangement)	0.438	4650044	0.288	Hobby (hill climbing)	0.481	4580916	0.215	Annual income for the entire household (200–250 thousand)
0.402	4610021	0.329	Hobby (Gym・Tai Chi)	0.425	4680029	0.265	Hobby(Go・Shogi・Mahjong)	0.449	0711454	0.206	Frequency of meeting with friends (1–3 per year)
0.393	4650074	0.321	What is the region for you (etc.)	0.423	4670062	0.255	Hobby(Photography)	0.446	4570813	0.206	Number of friends met last month (1–2 person)
0.389	4770035	0.310	Hobby (Haiku・Tanka・Poetry)	0.417	4640856	0.237	Drinking companion (wife・husband)	0.419	4610031	0.201	Frequency of participation in hobby groups (None)
0.369	4660033	0.309	Dinner companion (friends)	0.403	4650027	0.189	Drinking frequency (3–4 per week、1–3 per month)	0.408	4630803	0.196	Hobby (Fishing)

Figure 7: Postal codes and Question answers with the top 10 highest affiliation probabilities

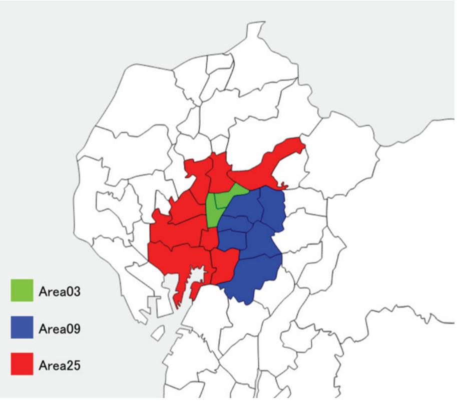


Figure 8: Cluster distribution in Nagoya city

dency of answers is similar when the residential areas are located nearby. There is a high possibility that there is a correlation between the response tendency of the elderly and the resident area. The results of figure 7 and figure 8 suggest there is regional disparities in terms of social network.

Bayesian network

In this research, we used Bayonet (Motomura 2003) to create a Bayesian network. The determination of the graph structure of the Bayesian network can be determined by Greedy algorithm that searches for the optimal local tree for each child node. We build Bayesian network by the procedure 1) child nodes are defined, 2) candidate local trees are given for each child node, 3) conditional probability is determined for each local tree, 4) an optimal local tree is searched for Greedy for each child node. In the procedure of 4), when choosing a tree, we select the candidate set given in advance by the selection criterion (MDL, AIC) which takes into consideration the likelihood and the complexity of the model. Extracting regional questions out of all questions, we constructed a bayesian network with the answer and the variables belonging to the above three clusters in figure 9. This questions are about the change in their area in the past 3 years and about the environment within 1 km from their house. In this bayesian network, the upstream is occupied by neighboring facilities, it propagates to the change of the area, and eventually leads to cluster affiliation. This result suggests that neighborhood facilities affect regional change, regional change affect regional disparity. By executing probabilistic reasoning on this bayesian network, the influence

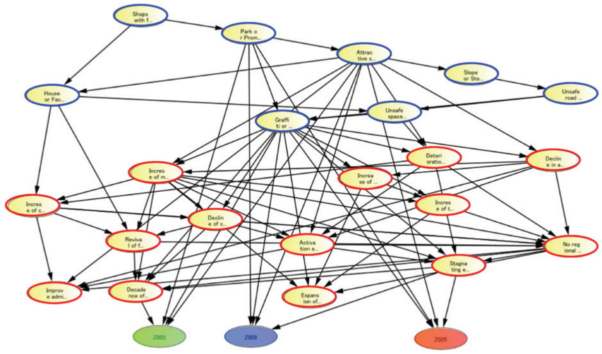


Figure 9: Bayesian Network about regional characteristics

degree between nodes can be quantitatively calculated.

We focus on the node of Increase of local communication and activity. There are three parent nodes linked to this node,"Houses and Facilities that you can drop in casually within 1 km from your house", "Increase of people moving in", "Deterioration in security". By giving numerical value into each node and comparing the prior probability with the posterior probability in figure 10, it increases from 3.7 percent to 19.8 percent by regional intervention. In other words, if you increase the number of homes and facilities that you can drop in easily, promote an increase in the number of people moving in and improve the public safety, it can be expected that the local community will be revitalized. As this inference on the top down, by giving numerical value to the parent node on the Bayesian network and looking at the probability transition of the child node, it is possible to deduce what kind of result will occur with a certain probability under a certain hypothesis.

On the other hand, by giving numerical value to a child node and looking at the probability transition of the parent node at the bottom up, the most likely hypothesis can be obtained when the result is given. As shown in figure 11, when the result of No in "Expansion of income gap" is obtained, the probability of Sometimes in "Graffiti or Garbage" decreased. Moreover, the probability of No in "Increase of unwaged" and "Increase of people moving in" has increased. This result suggests graffiti and garbage in the neighborhood and the increase of unwaged and people moving in are factors of expansion of income gap. In this way, it is possible to use the Bayesian network as a hypothesis construction.

Discussion

Importance of social capital

This study shows the presence of disparity of social network. Epidemiological studies have concluded that people who are socially integrated live longer(House, James S et al. 1988). Berkman's study have shown that the people who lacked social and community ties were more likely to die than those with more extensive contacts.

The age-adjusted relative risks for those most isolated when compared to those with the most social contacts were 2.3 for men and 2.8 for women(Berkman, Lisa F et al.1979).

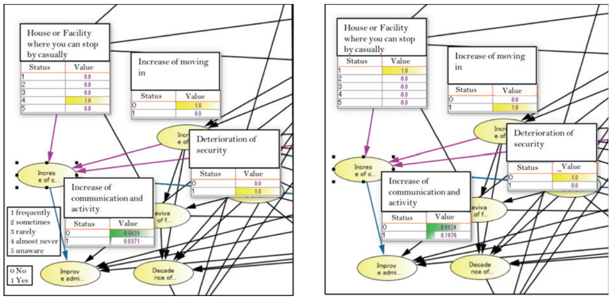


Figure 10: Bayesian Network Infer : Top-Down

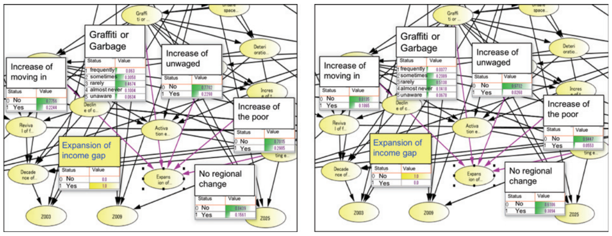


Figure 11: Bayesian Network Infer : Bottom-Up

These studies show the evidence that the social cohesion enhances longevity.Robert Putnam defined social capital as features of social organization, such as trust, norms, and networks, that can improve the efficiency of society by facilitating coordinated actions(Putnam, Robert D et al. 1994).

There are three plausible reasons why social capital affect individual health(Kawachi et al 1999; Kawachi et al. 1997; Kawachi et al. 1997). According to Kawachi's study, (1) social capital may influence the health behaviors of neighborhood residents by promoting more rapid diffusion of health information, increasing the likelihood that healthy norms of behavior are adopted (e.g., physical activity), and exerting social control over deviant health-related behavior, (2)neighborhood social capital may influence health by increasing access to local services and amenities, (3) neighborhood social capital may influence the health of individuals via psychosocial processes, by providing affective support and acting as the source of self-esteem and mutual respect.

In Japan, we always hear news about dying alone and social isolation of the elderly. On the other hand, Japan also have communities where there are strong connections through neighborhood associations and local governments. Local governments have the potential to greatly contribute to Japan's aging society by focusing on social capital. As part of that, this study suggest the method for regional characteristics extraction and decision support for regional intervention and showed the analysis results.

Social capital and Regional environment

From the view of social capital, there is hypothesis that explains the results of this study. From figure 7 and figure 8, the east side of the city is Z009, where the cluster has good environment, and seem to be no isolation because drinking with companion. On the other hand, the west side of the city is Z025. This cluster is the area where people lacked social and community ties. Nagoya city has large scale green parks with forest in the east side. For example, there are Higashiyama park and Heiwa park in Tikusa ward, Obata green tract of land in Moriyama ward, Makinogaike green tract of land in Meito ward and Odaka green tract of land in Midori ward. All of these parks are in the east side of Nagoya. The core of the city is the central part of Nagoya, where commercial facilities and office buildings were build. With figure 9, three cluster: Z003, Z009, Z025 have common parent node in the most upstream, it is just "Parks and promenades suitable for exercise and walking." Here is the hypothesis that the fact that there are parks in the neighborhood gives rise to social capital disparity. Ariane L's study shows psychological benefits for park users that arise from the proximity of natural environments(Bedimo-Rung, Ariane L et al. 2005). Other study (Godbey, Geoffrey, and Michael Blazey 1983) has shown that older adult park users who participated in moderate aerobic activity were in a better mood after visiting the park. From the perspective of public health, it would be beneficial to add parks and encourage social capital.

Policy Decision Support System in Aging Society

Gerontechnology is defined as interdisciplinary academic and professional field combining gerontology and technology(Bouma, Herman et al. 1992). This field not only supports the elderly but also will develop into a core industry in Japan. In China, the number of elderly people will exceed 200 million in 2025. Japan is expected to help asian countries and promotion of economic growth by gerontechnology.

Policy problems in aging society are also needs solution by Gerontechnology. Policy problems are complex structures involving various factors so that there is high possibility of misjudging the problem in the conventional field-specific ways. However, PLSS Modeling reflects the complexity of policy problems. As stated in introduction, there are difficulties in policy problems: 1)Comprehensiveness; BN capture the relationship of many variables without being bound by field-specific views. Moreover, BN reflects knowledge from multiple viewpoints and experts as prior distribution. 3)Subjectivity; The relationship between variables is visually clear and we share the frame of problems. 4)Dynamics: BN respond flexibly to new data and changes of policy problems. PLSS Modeling based on JAGES data enable local governments to construct hypothesis about values and needs of elderly people. Longitudinal data integration and analysis will improve prediction accuracy. JAGES aims to make smart aging society by artificial intelligence and develop health care simulation science. Figure 12 shows the framework of data platform in aging society. PHR means Personal Health Record, which is a collection of health-related information that is documented and maintained by the individual it pertains to. It is necessary to constantly reflect social feedback to the model without separating modeling and application using model. Therefore, we should follow the cycle as Figure 13. By continuing this cycle, it is expected that system will be established to continuously calculate the characteristics of diverse elderly people and utilize them as useful knowledge for society. Sharing this system will help realization of new social infrastructure in aging society.

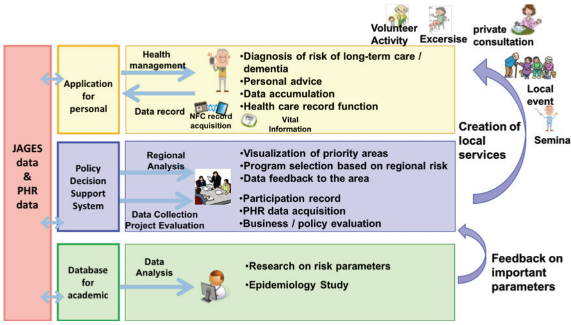

Volunteer Activity
Excersise
private consultation
JAGES data & PHR data
Application for personal
Health management
Data record
NFC record acquisition
•Diagnosis of risk of long-term care / dementia
•Personal advice
•Data accumulation
•Health care record function
Vital Information
Local event
Creation of local services
Seminar
Policy Decision Support System
Regional Analysis
Data Collection Project Evaluation
•Visualization of priority areas
•Program selection based on regional risk
•Data feedback to the area
•Participation record
•PHR data acquisition
•Business / policy evaluation
Database for academic
Data Analysis
•Research on risk parameters
•Epidemiology Study
Feedback on important parameters

Figure 12: JAGES and PHR data platform in aging society

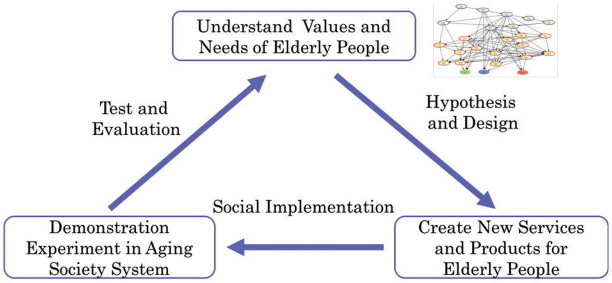

Understand Values and Needs of Elderly People
Hypothesis and Design
Create New Services and Products for Elderly People
Social Implementation
Demonstration Experiment in Aging Society System
Test and Evaluation

Figure 13: Cycle for solving problems in aging society

Conclusion

The paper has described Probabilistic Latent Spatial Semantic (PLSS) Modeling which is integrated probabilistic latent semantic analysis and bayesian network to uncover the current geographical status of living activities of elderly people. Finally, this paper proposed policy decision support system to implement PLSS Modeling in the real world.

We have clarified the latent regional characteristics and the factors of regional disparities from the elderly questionnaire data. We also mentioned what kind of intervention the municipal government should take to solve regional disparities. It became clear that 1)there is regional disparities in terms of social network, 2)neighborhood facilities affect regional disparity, 3)the local community will be revitalized by increasing the number of homes and facilities that you can drop in easily, promoting an increase in the number of people moving in and improving the public safety.

In the future, there is need to comparative controlled study to certify causality. By using the results of clustering regions, we are able to compare regions without intervention and with intervention in the same cluster. JAGES is engaged in intervention by holding local events. We plan health tracking such as blood pressure measurement at these events for model evaluation.

Our remaining works include i) undersampling data before constructing BN to improve the accuracy of prediction of health status; ii) time series data analysis using other year's survey data; iii) intervention trial to certify causality; These future works might help local government to improve social capital and health status.

Acknowledgment

In this research, we used the data of the Japan Gerontological Evaluation Study (JAGES) project. We would like to thank Dr. K Kondo and Dr. N Kondo for cooperating the launch of the artificial intelligence working group, and many people involved in the JAGES project such as Dr. T Tsuji , who provided dataset combined the postal code. In addition, this research was supported by NEDO entrusted project "Research and development of next generation artificial intelligence technology that can mutually understand with human beings."

References

Akaike, Hirotugu. Factor analysis and AIC. Psychometrika 52.3 (1987): 317-332.

Bedimo-Rung, Ariane L., Andrew J. Mowen, and Deborah A. Cohen. ”The significance of parks to physical activity and public health: a conceptual model." American journal of preventive medicine 28.2 (2005): 159-168.

Berkman, Lisa F., and S. Leonard Syme. Social networks, host resistance, and mortality: a nine-year follow-up study of Alameda County residents. American journal of Epidemiology 109.2 (1979): 186-204.

Bouma, Herman, and Jan AM Graafmans, eds. Gerontechnology. Vol. 3. IOS Press, 1992.

Godbey, Geoffrey, and Michael Blazey. ”Old people in urban parks: an exploratory investigation." Journal of Leisure Research 15.3 (1983): 229.

Hirokawa Noriaki, Keisuke Murayama and Yoichi Motomura. Probabilistic Latent Spatiotemporal Semantic Structure Models Based on Travel History Data for Regional Revitalization. ICserv 2015, 2015

Hofmann, Thomas. Probabilistic latent semantic indexing. Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1999.

House, James S., Karl R. Landis, and Debra Umberson. Social relationships and health. Science 241.4865 (1988):540.

Ishigaki, Tsukasa, Takeshi Takenaka, and Yoichi Motomura. Category mining by heterogeneous data fusion using PdLSI model in a retail service. Data Mining (ICDM), 2010 IEEE 10th International Conference on. IEEE, 2010.

Kawachi, Ichiro, Bruce P. Kennedy, and Roberta Glass. Social capital and self-rated health: a contextual analysis. American journal of public health 89.8 (1999): 1187-1193.

Kawachi, Ichiro, and Bruce P. Kennedy. Health and social cohesion: why care about income inequality?. BMJ: British Medical Journal 314.7086 (1997): 1037.

Kawachi, Ichiro, et al. Social capital, income inequality, and mortality. American journal of public health 87.9 (1997): 1491-1498.

Motomura, Yoichi. Predictive modeling of everyday behavior from large-scale data. Synthesiology English edition 2.1 (2009): 1-12.

Motomura,Yoichi. Bayesian Network Software Bayonet, Journal of the Society of Instrument and Control Engineers, 2003,pp.693-694

Murayama, Keisuke, Noriaki Hirokawa, and Yoichi Motomura. Customer Behavior Analysis Using Probabilistic Latent Structure Modelling with Point Card Data. ICserv 2015, 2015.

Pearl, Judea. Bayesian networks: A model of self-activated memory for evidential reasoning. Proceedings of the 7th Conference of the Cognitive Science Society 1985,1985.

Pfeiffer, Dirk, et al. Spatial analysis in epidemiology. Vol. 142. New York, NY, USA:: Oxford University Press, 2008.

Putnam, Robert D., Robert Leonardi, and Raffaella Y. Nanetti. Making democracy work: Civic traditions in modern Italy. Princeton university press, 1994.

Snow, John. On the mode of communication of cholera. John Churchill, 1855.

Takao Akiyoshi, Shuichiro Ito, and Toshiya Kitayama. Foundations of Public Policy Studies.,Yuhikaku books, 2015.

Texture Suggestion System Considering the Elderly's Preference on 3D Printing

Tomohiko Inazumi,[†] Jinhwan Kwon,[†] Shinsaku Hiura,[††] Maki Sakamoto[†]

[†]The University of Electro-Communications, Tokyo, Japan
[††]Hiroshima City University, Hiroshima, Japan
i1410011@edu.cc.uec.ac.jp, {kwonjh, maki.sakamoto}@uec.ac.jp, hiura@hiroshima-cu.ac.jp

Abstract

Changes in sensation due to aging affect the preferences when the elderly choose products. Texture is an important factor to affect their preferences of products especially in the case of items that are held by hand. However, it is difficult to design products preferred by elderly people. Therefore, we constructed a texture suggestion system considering the elderly's preference. Users input sound-symbolic words such as "tsuru-tsuru" (comfortable slippery texture), a kind of Japanese intuitive texture word. Then the system refers to the texture database including 3D model images outputs 3D model data to realize the texture design preferred by elderly people.

Introduction

Technological development of 3D printers is remarkable. It is possible to manufacture microscopic and fine objects of micron size. In addition, there is also great convenience that various materials such as resin and metal can be used as materials (Bhushan and Caspers 2017). The printing technology is highly applicable because we can use many different materials. It is also possible to make artificial body parts to create various products. There are few studies considering individual texture preference, especially preference of elderly people. In this study, we focus on the additive manufacturing techniques (3D printers) to consider the relationship between texture property and preference of elderly people.

We can perceive surface textures and material properties through the sense of touch, and the tactile modality is considered to play an important role in evaluations of items used in daily life. In particular, relations between surface's material properties and tactile perception are important for consumer goods. In the psychophysical domain, a large body of literature deals with the relationship between surface (or texture) perceptions and the surfaces' physical properties (Bergmann Tiest and Kappers 2006, Chen et al. 2009, Gescheider et al. 2005, Hollins et al. 2000, Hollins et al. 1993, Okamoto, Nagano and Yamada 2013, Picard et al. 2003). However, most of them have identified the main factors of tactile perception based on averaged data among subjects (Hollins et al. 2000). Touch quality is influenced not only by the surfaces' physical properties of the object but also by a large individual difference such as skin deformation of the finger.

In recent years, many researchers have studied sound symbolism as an integral expression of texture, and have verified its effectiveness (Doizaki, Watanabe and Sakamoto 2017, Sakamoto and Watanabe 2016, Sakamoto and Watanabe 2017, Sakamoto et al. 2016). Sound symbolism represents a phenomenon in which a certain amount of information perceived from environment is strongly associated with phonological elements (as sound) in the brain. The existence of sound symbolic words (hereafter, SSWs) has been demonstrated in a wide variety of languages (Bolinger 1950, Hinton, Nichols and Ohala 1994, Köhler 1929, Nuckolls 1999, Ramachandran and Hubbard 2001, Sapir 1929, Schmidtke, Conrad and Jacobs 2014). For example, English words starting with "sl-" such as "slime", "slush", "slop", "slobber", "slip", and "slide" symbolize something smooth or wet (Bloomfield 1933). We have proposed a system that can automatically estimate multi-dimensional ratings of touch from a single sound-symbolic word that has been spontaneously and intuitively expressed by a user (Ramachandran and Hubbard 2001). When a user inputs a sound-symbolic word into the system, the system refers to a database of phonemes and their auditory impressions, and calculates ratings in terms of 26 pairs of fundamental scales of touch. The estimated ratings of sound-symbolic words enable us to visualize a tactile perceptual space. We assume that the human brain has a database of phonemes and perceptual learning and SSW can be used for integral expression of texture.

Aging can be regarded as an important factor of individual difference in touch because functions in the main sensory modalities are declined with aging. In particular, the effect of aging on touch is well reported in nervous system, tactile thresholds and functional implications (Nusbaum. 1999, Wickremaratchi and Llewelyn 2006). Aging leads to a gradual decrease of cells and fibers in the central and peripheral nervous system (Katzman and Terry 1983, McLeod 1980, Victor and Ropper 2001). The density and distribution of Pacinian and Meissner corpuscles and Merkel's discs also decrease in the skin of the elderly people (Bolton, Winkelmann and Dyck PJ 1966, Gescheider et al. 1994, Schimrigk and Ruttinger 1980, Stevens and Patterson 1995). In addition, older subjects were significantly less sensitive to mechanical stimuli and tactile, vibration, pain and temperature thresholds in the elderly are significantly increased (Gescheider et al. 1996, Gescheider et al. 1994, Goble, Collins and Cholewiak 1996, Kenshalo 1986, Thornbury and Mistretta 1981, Tucker et al. 1989, Verrillo, Bolanowski and Gescheider 2002). Thus, aging in tactile sense can be an important issue. With respect to that issue, it is necessary to propose methods and systems to reflect in detail the differences in touch feeling of individuals which was difficult to quantitatively assess.

There are several studies suggesting the possibility that changes in sensation may affect preferences. Kuga (Kuga 1996) suggests that threshold changes tend to affect preferences and pointed out a change in taste preference. Furthermore, there are several studies related to texture and preference, for example, the relation between the perception and discomfort (Iwasa and Komatsu 2015) and, the preferable texture on touch (Mihara, Sekine and Yamauchi 2007). However, there are very few studies focusing on the preferred texture for the elderly. When comparing young people and elderly people, texture preferences may be different. Since the population of elderly people is increasing, it will be getting more important to consider the preference of elderly people. Still, it seems that the most products of everyday life are designed for young people because designers of products are relatively young. Therefore, there is a need to create a system to recommend texture design preferred by the elderly.

The purpose of this research is to construct texture suggestion system considering the elderly's preference on 3D modeling. To achieve the purpose, we investigate the tendency of texture preference of the elderly and construct the system using the analysis results.

System Construction

In this research, we construct a system that proposes 3D texture model to consider preferences of elderly people. Users input a SSW to express intuitively the texture they want to be put on their products. Then, the system refers to the texture database including 3D model image and designs are outputted from the 3D model. In this research, we focus on shape as texture parameter. First, we selected SSWs for input data and constructed a database that store texture data corresponding to SSWs. In particular, the texture data shows the 3D model using surface characteristics (height, width and interval). Figure 1 shows the flow of the system. The overview of the operation of the system is as follows, (1) Enter the SSW that matches the texture you want. (2) The system refers to the database and call the 3D model data corresponding to the input SSW. (3) The system outputs using the 3D model data.

Selection of input SSWs

First, we perform an experiment to select input SSWs. 60 elderly participants participated in the experiment. In the experiment, 18 types of image data were used and the texture was evaluated by SSWs and semantic differential (SD) method. The experimental stimuli used in this study were obtained from the FMD (http://people.csail.mit.edu/celiu/CVPR2010/FMD/) (Sharan et al. 2014), which is one of the major stimulus sets used in vision research. The FMD consists of color photographs of surfaces belonging to one of ten common material categories: fabric, foliage, glass, leather, metal, paper, plastic, stone, water, and wood.

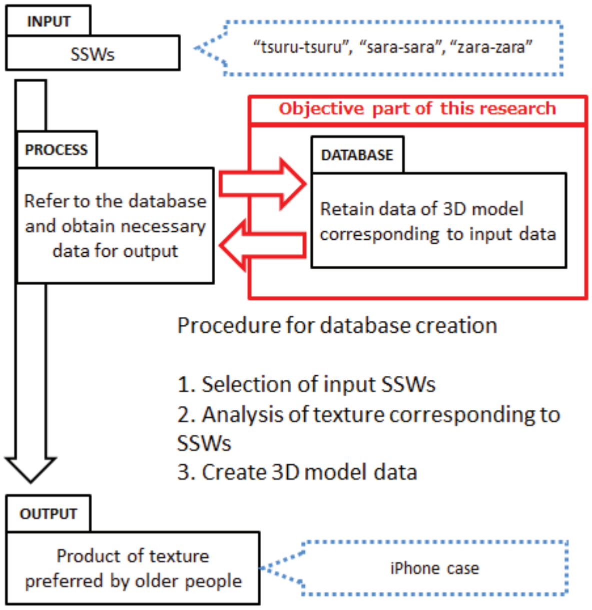

INPUT
SSWs
"tsuru-tsuru", "sara-sara", "zara-zara"
Objective part of this research
PROCESS
Refer to the database and obtain necessary data for output
DATABASE
Retain data of 3D model corresponding to input data
Procedure for database creation
1. Selection of input SSWs
2. Analysis of texture corresponding to SSWs
3. Create 3D model data
OUTPUT
Product of texture preferred by older people
iPhone case

Figure1. Flow of system that we aim for.

Each image contains surfaces that belong to a single material in the foreground. A range of images was selected to provide a variety of illumination conditions, compositions, colors, textures, surface shapes, material sub-types, and object associations. Since the FMD was constructed with the specific goal of capturing the natural range of material appearances, the surfaces depicted in the images each belong to a specific material category, and not any of the others. In the FMD images, we selected 18 types of image data as experimental stimuli representing the basic 6 tactile scales, "warm - cool", "hard - soft", "slippery - sticky", "dry - wet", "coarse - smooth", "uneven - flat".

The experimental method is as follows. (1) The selected image stimulus is presented to the participants. (2)Participants answered whether they want to touch those appearing in the photograph by using 7-step SD method. (3) Participants were instructed to answer spontaneously and freely SSWs expressing the texture of each material.

From the result of the answer, “Zara-Zara”, “Sara-Sara” and “Tsuru-Tsuru” were most frequently answered with preference. Therefore, we selected them as input data.

Database construction

Data stored in database shows the 3D model images created by texture data corresponding to input data and the data is used when outputting by a 3D printer. Therefore, we conducted the experiment to obtain data on the surface shape corresponding to SSWs. At first, we prepared the tactile stimuli to investigate the texture pattern of surface characteristics which are height, width and interval. 40 kinds of texture stimuli are provided by Nice and Takeo Co. as experimental stimuli. We analyzed surface characteristics (height, width and interval) using a 3D surface measurement system (KEYENCE/VR-3000). Figure 2 shows the definition of surface characteristics which are height, width and interval.

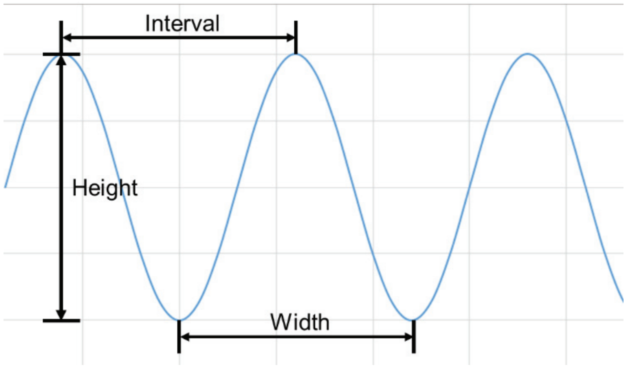


Figure 2. Surface characteristics which are height, width and interval.

30 participants participated in the experiment. The experimental method is as follows. (1) Participant touches the surface of the experimental stimulus. (2) Participants were instructed to answer spontaneously and freely with SSWs expressing the texture of each material.

We analyzed the height and width of the experimental stimuli corresponding to "Zara-Zara", “Sara-Sara” and “Tsuru-Tsuru,” which are frequently answered. From the analysis, we obtained the mean values of height and width. Then, we created the 3D model using the shape’s data and we used the 3D modeling data for outputting the objects.

Output object

Next, we apply our shape data to output object. We decided a smartphone case as output object because smart phones and iPhones are spread to all generations. We used Fusion 360 (3D CAD software) to create 3D modeling data to create 3D modeling data. First, 3D modeling data of the texture was created based on "height" and "width" corresponding to SSWs obtained from the experimental result. Specifically, we created texture data repeatedly using the waveform of the height and width. Figure 3 shows the example of 3D model of iPhone case. Then we stored the 3D modeling data in the database. With respect to 3D printing, 3D printer (Stratasys Objet 260 Connex 3) was used to output the 3D modeling data. The output mode was matte mode, VeroBlackPlus (black opaque hard resin) was used as the material.

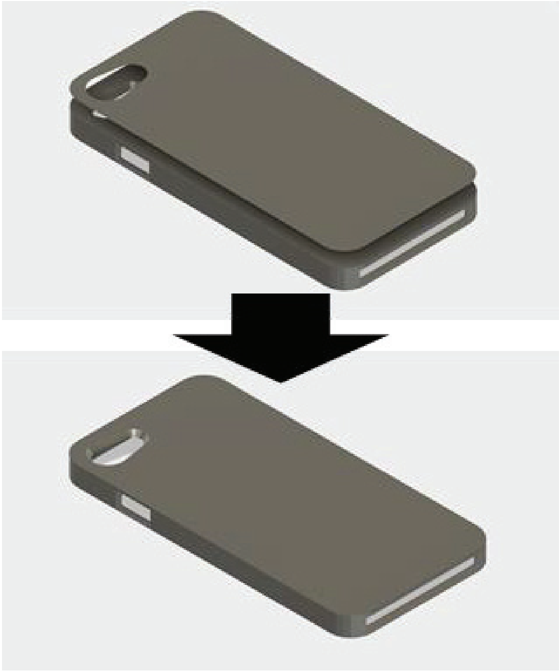
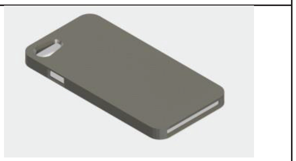
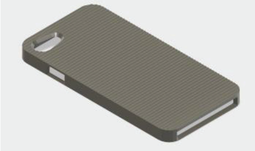
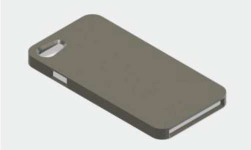
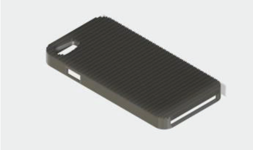
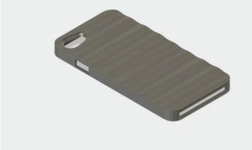
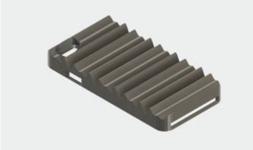


Figure 3. The example of 3D model of iPhone case.

Evaluation of the outputs

We printed the iPhone case using the data in the database. However, we need to confirm whether these iPhone cases are preferred by the elderly. Therefore, we additionally printed the comparable iPhone cases. The comparable iPhone cases were targeted at iPhone cases that were fivetimes larger than the shape of the 3D model in the database, random shapes, and no textures (flat surface). Table 1 shows the outputs of the iPhone case. 40 elderly participants

Table 1. The experimental stimuli

No	Name	Height[mm]	Width[mm]	Output image
1	Zara-Zara	0.079±0.083	0.482±0.195	
2	Sara-Sara	0.021±0.017	0.345±0.121	
3	Tsuru-Tsuru	0.005±0.002	0.222±0.041	
4	Enraged shape from the No1 stimulus	0.393±0.416	2.411±0.974	
5	Enraged shape from the No2 stimulus	0.105±0.086	01.727±0.603	
6	Enraged shape from the No3 stimulus	0.023±0.011	1.109±0.203	
7	Standard stimulus	0	0	
8	Random shape 1	5.000		
9	Random shape 2	3.220		
10	Random shape 3	10.400		

participated in the experiment. This experiment was conducted to confirm whether the output objects are preferred or not. The procedure is as follows. (1) Participants touched the ten stimuli, and were instructed to evaluate and answer the texture of each stimulus using three SSWs, "Zara-Zara", "Sara-Sara" and "Tsuru-Tsuru". (2) Participants evaluated whether they prefer the output's textures using 7-step SD method. iPhone mockup was used in this experiment to more realistically evaluate. Figure 4 shows the experimental situation.

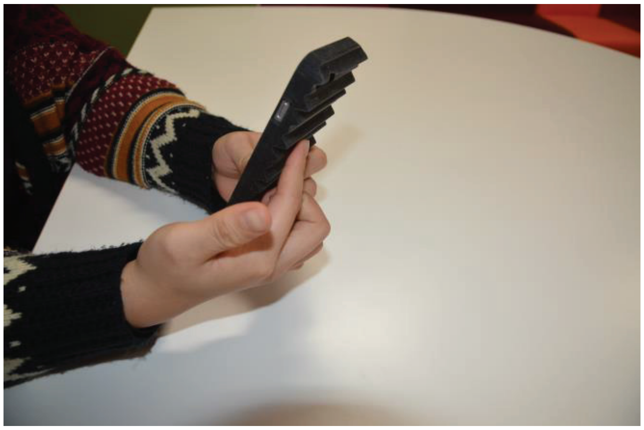


Figure 4.Experimental situation

Result

Table 2 shows the result of experiment. Participants rated the stimuli 1 to 7 as "preferred" and the stimuli 8 to 10 as "non-preferred". We confirmed whether the iPhone cases outputted from the database were preferred or not and were answered to correspond with database's SSWs. As a result, although stimulus No. 3 corresponding to "Tsuru-Tsuru" was answered as database's data, but Stimuli No. 1 and 2 were not answered as database's SSWs.

Then, we performed comparative analysis between iPhone cases outputted from the database and those that were not. At first, we classified experimental stimuli based on SSW's answers. As a result, the stimuli No. 4, 8 and 10 were frequently answered as "Zara-Zara", the stimuli No. 1, 6 and 9 were frequently answered as "Sara-Sara" and the stimuli No. 2 and 3 were frequently answered as "Tsuru-Tsuru", respectively. In "Tsuru-Tsuru" group, a repeated-measures analysis of variance (ANOVA) of the preferences did not show a significant main effect for the shape data (p=.275). In "Sara-Sara" group, a repeated-measures ANOVA of the preferences showed a significant main effect for the shape data (p<.001). Therefore, Stimulus 1 is significantly more preferred than other stimuli. Furthermore, in "Zara-Zara" group, a repeated-measures ANOVA of the preferences showed a significant main effect for the shape data (p<.001). Therefore, Stimulus 4 is significantly more preferred than other stimuli.

Table 2. The result of experiment

No	Frequently Answered SSWs	Degree of Preference (mean value)
1	Sara-Sara	1.250
2	Tsuru-Tsuru	1.2575
3	Tsuru-Tsuru	1.375
4	Zara-Zara	0.425
5	Zara-Zara	0.500
6	Sara-Sara	0.425
7	Tsuru-Tsuru	1.050
8	Zara-Zara	-0.450
9	Sara-Sara	-0.125
10	Zara-Zara	-1.750

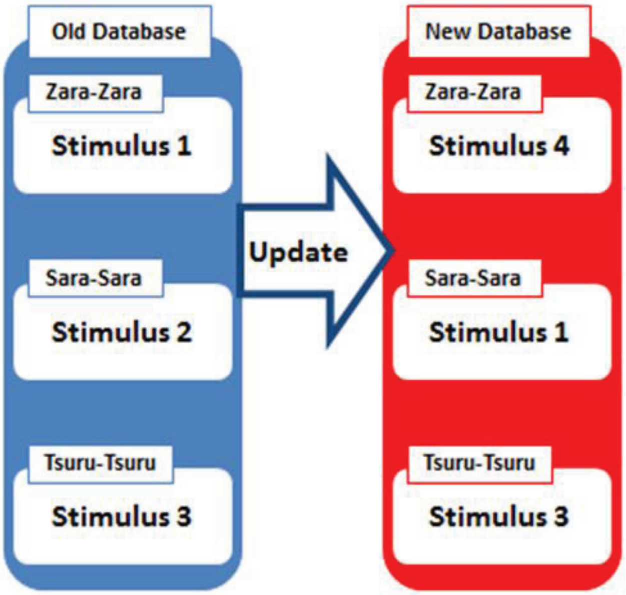


Figure 5. Updated Database

Based on the results, the original data (stimulus 3) corresponding to "Tsuru-Tsuru" can be used to output the object. However, there is a need to change the original data of stimuli 1 and 2. Therefore, we updated the 3D modeling data based on the results. Specifically, the 3D modeling data corresponding to "Zara-Zara" changed from the stimulus 1 to the stimulus 4 and the 3D modeling data corresponding to "Sara-Sara" changed from the stimulus 2 to the stimulus 1. Figure 5 shows the update of database.

Conclusion

In this study, we constructed a texture suggestion system considering the elderly's preference. Users input sound-symbolic words such as "tsuru-tsuru" (comfortable slippery texture), a kind of Japanese intuitive texture word.

Then the system refers to the texture database including 3D model images outputs 3D model data to realize the texture design preferred by elderly people. In future work, we aim to construct a system that can handle not only three SSWs for input but also all SSWs, for example, using a method such as focusing on phonemes and structure. In addition, the database in this study focused only on the surface shape of the object. Therefore, we aim to use data of other physical quantity forming texture such as hardness, moisture.

Acknowledgements

This work was supported by Grant-in-Aid for Scientific Research (A) Grant Number 15H01720) and JSPS KAKENHI Grant Number 15H05922 (Grant-in-Aid for Scientific Research on Innovative Areas "Innovative SHITSUKSAN Science and Technology")from MEXT, Japan.

References

Bergmann Tiest, W.M.; and Kappers, A.M.L. 2006. Analysis of haptic percep-tion of materials by multidimensional scaling and physical measurements of roughness and compressibility. *Acta Psychologica* 121(1):1-20.

Bhushan, B.; and Caspers, M. 2017. An overview of additive manufacturing (3D printing) for microfabrication. *Microsystem Technologies archive* 23(4)1117-1124.

Bloomfield, L. 1933. Language. New York, NY: Henry Holt.

Bolinger, D. 1950. Rime, assonance, and morpheme analysis. Word 6:117–136.

Bolton, C.F.; Winkelmann, R.K.; and Dyck PJ, P.J. 1966. A quantitative study of Meissner's corpuscles in man. *Neurology* 16(1):1–9.

Chen, X.; Shao, F.; Barnes, C.; Childs, T.; and Henson, B. 2009. Exploring relationships between touch perception and surface physical properties. *International Journal of Design* 3(2):67-77.

Doizaki, R.; Watanabe, J.; and Sakamoto, M. 2014. A System for Evaluating Tactile Feelings Expressed by Sound Symbolic Words. M. Auvray and C. Duriez (Eds.): In *EuroHaptics* 2014 Proceedings Part I, LNCS 8618, 32-39. Springer, Heidelberg.

Doizaki, R.; Watanabe, J.; and Sakamoto, M. 2017. Automatic Estimation of Multidimensional Ratings from a Single Sound-symbolic Word and Word-based Visualization of Tactile Perceptual Space. *IEEE Transaction on Haptics* 10(2):173-182.

Gescheider, G.A.; Beiles, E.J.; Checkosky, C.M.; Bolanowski, S.J.; and Verrillo, R.T. 1994. The effects of ageing on information processing channels in the sense of touch: II. Temporal summation in the P channel. *Somatosensory & Motor Research* 11(4):359–365.

Gescheider, G.A.; Bolanowski, S.J.; Greenfield, T.C.; and Brunette, K.E. 2005. Perception of the tactile texture of raised-dot patterns: A multidimensional analysis. 22(3):127-140.

Gescheider, G.A.; Bolanowski, S.J.; Hall, K.L.; Hoffman, K.E.; and Verrillo. R.T. 1994. The effects of ageing on information-processing channels in the sense of touch: I. Absolute sensitivity. *Somatosensory & Motor Research* 11(4)345–357.

Gescheider, G.A.; Edwards, R.R.; Lackner, E.A.; Bolanowski, S.J.; and Verrillo, R.T. 1996. The effects of ageing on information-processing channels in the sense of touch: III. Differential sensitivity to changes in stimulus intensity. *Somatosensory & Motor Research* 13(1):73–80.

Goble, A.K.; Collins, A.A.; and Cholewiak, R.W. 1996. Vibrotactile threshold in young and old observers: the effect of spatial summation and the presence of a rigid surround. *The Journal of the Acoustical Society of America* 99(4):2256–2269.

Hinton, L.; Nichols, J.; and Ohala, J. eds. 1994. Sound Symbolism. Cambridge: Cambridge University Press.

Hollins, M.; Bensmaïa, S.; Karlof, K.; and Young, F. 2000. Individual differences in perceptual space for tactile textures: Evidence from multidimensional scaling. Attention. *Perception & Psychophysics* 62(8):1534-1544.

Hollins, M.; Faldowski, R.; Rao, S.; and Young, F. 1993. Perceptual dimensions of tactile surface texture: A multidimensional scaling analysis. *Perception & Psychophysics* 54(6):697-705.

Iwasa, K.; and Komatsu, T. 2015. Influence of naming on visual touch sensation perception and discomfort. *The Japanese Society for Artificial Intelligence* 30(1):265-273.

Katzman, R.; and Terry, R.D. 1983. Normal ageing of the nervous system. Neurology of ageing, Katzman, R.; and Terry, R.D. eds., Philadelphia: Davies:15–50.

Nuckolls, J. 1999. The case for sound symbolism. *Annual Review of Anthropology* 28:225–252.

Kenshalo, D.R. 1986. Somesthetic sensitivity in young and elderly humans. *The Journals of Gerontology* 41(6):732–742.

Köhler, W. 1929. Gestalt Psychology. NewYork, NY: Liveright Publishing Corporation.

Kosahara, M.; Watanabe, j.; Hiranuma, Y; Doizaki, R.; Matsuda, T.; and Sakamoto, M. 2016. A System to Visualize Tactile Perceptual Space of Young and Old People., In *AAAI Spring Symposium Series*, 375-380. Stanford University, Calif.: AAAI Publications.

Kuga, M. 1996. A study on changes in taste function due to pregnancy. *Proceedings of the Japan Otolaryngology Society* 99(9):1208-1217.

McLeod, J.G. 1980. The effect of ageing on the neuromuscular system of man: a review. *Aging Clinical and Experimental Gerontology* 2(1):259-269.

Nusbaum, N.J. 1999. Ageing and sensory senescence. *The Southern Medical Journal* 92(3):267–75.

Okamoto, S.; Nagano, H.; and Yamada, Y. 2013. Psychophysical Dimensions of Tactile Perception of Textures. *IEEE Transactions on Haptics* 6(1):81-93.

Picard, D.; Dacremont, C.; Valentin, D.; and Giboreau, A. 2003. Perceptual di-mensions of tactile textures. *Acta Psychologica* 114(2):165-184.

Sakamoto, M.; and Watanabe, J. 2017. Exploring Tactile Perceptual Dimensions Using Materials Associated with Sensory Vocabulary. *Frontiers in Psychology* 8:1-10.

Sekine, M.; Mihara, I.; and Yamauchi, Y. 2007. High-Dimensional Texture Technology for Photorealistic Computer Graphics. *TOSHIBA REVIEW* 62 (12):22-25.

Ramachandran, V.S.; and Hubbard, E.M. 2001. Synaesthesia—A window into perception, thought and language. *Journal of Consciousness Studies* 8(12):3–34.

Sakamoto, M.; Yoshino, J.; Doizaki, R.; and Haginoya, M. 2016. Metal-like Texture Design Evaluation Using Sound Symbolic

Words. *International Journal of Design Creativity and Innovation* 4(3-4):181–194.

Sakamoto, M.; and Watanabe, J. 2016. Cross-Modal Associations between Sounds and Drink Tastes/Textures: A Study with Spontaneous Production of Sound-Symbolic Words. *Chemical Senses* 41(3):197-203.

Sapir, E. 1929. A study in phonetic symbolism: *Journal of Experimental Psychology* 12:225–239.

Schmidtke, D.S.; Conrad, M.; and Jacobs, A.M. 2014. Phonological iconicity. *Frontiers in Psychology.* 5(80).

Schimrigk, K.; and Ruttinger, H. 1980. The touch corpuscles of the plantar surface of the big toe. Histological and histometrical investigations with respect to age. *European Neurology* 19(1): 49–60.

Stevens, J.C.; and Patterson, M.Q. 1995. Dimensions of spatial acuity in the touch sense: changes over a life span. *Somatosensory & Motor Research* 12(1):29–47.

Thornbury, J.M.; and Mistretta, C.M. 1981. Tactile sensitivity as a function of age. *Journal of Gerontology* 36(1): 34–39.

Tucker, M.A.; Andrew, M.F.; Ogle, S.J.; and Davison, J.G. 1989. Age associated change in pain threshold measured by transcutaneous neuronal electrical stimulation. *Age and Ageing* 18(4): 241–246.

Verrillo, R.T.; Bolanowski, S.J.; and Gescheider, G.A. 2002. Effects of ageing on the subjective magnitude of vibration: *Somatosensory & Motor Research* 19(3):238–244.

Victor, M.; and Ropper, A.H. 2001. The neurology of ageing. Principles of neurology. 7th ed., McGraw-Hill, 644–647.

Wickremaratchi, M.M.; and Llewelyn, J.G. 2006. Effects of ageing on touch. *Postgraduate Medical Journal* 82(967):301–304.

The Challenges for Understanding Cognitive Bias and Humanity for Well-Being AI — Beyond Machine Intelligence

Takashi Kido
Preferred Networks, Inc.
kido.takashi@gmail.com

Keiki Takadama
The University of Electro-Communications
keiki@inf.uec.ac.jp

Abstract

In this AAAI Spring symposium 2018, we discuss cognitive bias and humanity in the context of well-being AI. We define "well-being AI" as an AI research paradigm for promoting psychological well-being and maximizing human potential. The goals of well-being AI are (1) to understand how our digital experience affects our health and our quality of life and (2) to design well-being systems that put humans at the center. The important challenges of this research are how to quantify subjective things such as happiness, personal impressions, and personal values, and how to transform them into scientific representations with corresponding computational methods.

One of the important keywords in understanding machine intelligence in human health and wellness is cognitive bias. Advances in big data and machine learning should not overlook some new threats to enlightened thought, such as the recent trend of social media platforms and commercial recommendation systems being used to manipulate people's inherent cognitive bias.

The second important keyword is humanity. Rational thinking, on which early AI researchers had been focused their efforts, is recently and rapidly replacing human thinking by machines. Many people might have begun to believe that irrational thinking is the root of humanity. Empirical and philosophical discussions on AI and humanity would be welcome.

This paper describes the detailed motivation, technical, and philosophical challenges of this symposium proposal.

Motivation for Understanding Cognitive Bias and Humanity

Recent AI technologies (such as Deep Learning and other advanced machine learning technology) will definitely change the world. However, it seems that many people have excessive expectations or fears for AI in the near term, perhaps as a result of how AI is portrayed in science fiction and covered by the popular media. Examples include thinking that general purpose AI is just around the corner, as well as the fear the AI will steal jobs and create mass unemployment. It is therefore important that we first understand both the possibilities and limitations of the current machine intelligence correctly.

It remains especially challenging to understand the implications of machine intelligence in human health and wellness domains [Kido and Takadama, 2017]. Although statistical machine learning methods can predict the future based on past data, it remains difficult to respond to the new event which has never seen in the past. How to create new value that really makes people happy is one of the most important challenges in well-being AI. For this purpose, we need to share interdisciplinary scientific findings between human science (brain science, biomedical healthcare, psychology, and others) and AI.

One of the important keywords in this year's symposium is cognitive bias. In the recent trend of big data becoming more personalized, AI technologies to manipulate cognitive bias have evolved; For example, social media platforms such as Twitter and Facebook, and commercial recommendation system make it easy for people with the same opinion to form communities in which it appears that everyone has the same opinion; this is sometimes called the "Echo chamber effect." Recently, there has been a movement to use such cognitive bias in the political world as well. Advances in big data and machine learning should not overlook these new threats to enlightened thought.

The second important keyword in this symposium is humanity. One of the purposes of AI is to pursue "what is intelligence?" Early AI researchers focused their efforts to make progress on rational thinking, such as mathematical theorem proving, chess and so on. However, rational thinking is recently and rapidly being replaced by machines. It seems that many people might have begun to believe that irrational thinking is the root of humanity. Empirical and philosophical discussions on AI and humanity will be very important issues, if we design well-being AI systems that put human at the center.

Technical challenges and philosophical discussion

We need to deepen the understandings of the following four technical challenge areas of well-being AI as well as one philosophical issue. Technical research that clarifies the possibilities and limitations of “machine intelligence” or philosophical discussions on “AI and Humanity” are our important focuses.

(1) Representation of cognitive biases and personal traits.

First, we need to represent the cognitive biases, human tacit and subjective health/wellness knowledge in explicit and quantifiable ways. Much of the knowledge in well-being science is subjective. For example, fuzzy properties of subjective word embeddings in human health and wellness might be better represented with concrete mathematical structures.

(2) Representation of cognitive biases and personal traits.

Second, we need to explore the available advanced machine learning technologies, such as deep learning and other quantitative methods, in health and wellness domains. At the present, machine learning research is focused on giving machines the ability to recognize things and understand data similar to humans, such as recognizing images, text, sounds, and so on. However, the focus is going to shift to getting machines to understand things that humans cannot. We need to make a bridge to allow humans to understand these things.

(3) Interpretable Models, Reasoning and Inference

Third, the reasoning about data through representations should be understandable and accountable to human. For example, we need to develop powerful tools for understanding what exactly, deep neural networks and other quantitative methods are doing. Not only for increasing accuracy rate of predictions, we need to understand the causality with reliable models, reasoning and inference.

(4) Better Well-being systems design.

Furthermore, we need to understand the human. While recent technological advances bring many truly great benefits, there is an opportunity to rethink about the impact of these fruits. We need to understand how our AI revolution affects our emotions and our quality of life and how to design a better well-being system that puts humans at the center.

(5) Discussion on “AI and Humanity”.

We need to deepen the empirical and philosophical understandings on “AI and Humanity.” The topics include the Machine Intelligence vs. Human Intelligence”, or “How AI affects our human society or way of thinking”.

Conclusion

In this paper, we described the motivation, technical, and philosophical challenges related to “cognitive bias and humanity for well-being AI” as proposers and organizers of this AAAI18 symposium.

This symposium is aimed at sharing the latest progress, current challenges and potential applications related with AI, health, and well-being. The evaluation of digital experience and understanding of human health and well-being will be very important issues for designing human centric well-being AI.

References

Kido,T., Takadama, K. 2017. WELLBEING AI: FROM MACHINE LEARNING TO SUBJECTIVITY ORIENTED COMPUTING, AAAI *Spring symposium 2018* March, Stanford: https://aaai.org/Library/Symposia/Spring/ss17-08.php

Acknowledgments

I appreciate Dr. Hiroshi Maruyama at preferred Networks, Inc. The idea of focusing on “cognitive bias” for this workshop proposal was coming up with the valuable discussions with him.

Developing a Dataset for Personal Attacks and Other Indicators of Biases

John Licato
Advancing Machine and Human Reasoning (AMHR) Lab
Department of Computer Science and Engineering
University of South Florida
licato@usf.edu

Mark Boger and **Zhitian Zhang**
Department of Computer Science
Purdue University - Fort Wayne
bogem01@students.ipfw.edu
zhang820@purdue.edu

Abstract

Online argumentation, particularly on popular public discussion boards and social media, is rich with fallacy- and bias-prone arguments. An artificially intelligent tool capable of identifying potential biases in online argumentation might be able to address this growing problem, but what would it take to develop such a tool? In this paper, we attempt to answer this question by carefully defining both argumentative biases and fallacies, and laying out some guidelines for automated bias detection. After laying out a roadmap and identifying current bottlenecks, we take some initial steps towards relieving these limitations through the creation of a dataset of personal and *ad hominem* attacks in comments. Our progress in this direction is summarized.

Debates on Internet discussion boards are rarely, if ever, carried out in a calm, respectful, and open-minded manner by all participants. Exchanges carried out in the comments sections of websites like Facebook, Reddit, or YouTube will often devolve into the typed equivalent of shouting matches, rather than the high-minded exchange of ideas that one might hope to see more of. Many reasons have been put forward to explain this: the emergence of *trolling* (Cheng et al. 2017), accusations of government-sponsored agents intentionally fomenting discord (Woolley and Howard 2016), and so on.

There are some discussion forums which have been designed in such a way that encourages well-thought-out comments or content from relevant experts (e.g. StackExchange[1] or Reddit's AskScience subreddit,[2] even going so far as to aggressively remove sub-standard comments and ban users who are repeat offenders. However, such moderation is typically labor-intensive and can result in substantially decreased site activity, as significantly fewer users are able or willing to take the time to submit content meeting these increased standards.

We propose to develop automated tools to help with the problem of poor online discourse by addressing one of its primary causes: the prevalence of cognitive biases in online argumentation. In this paper, we describe first steps towards a research program whose goal is the development of algorithms capable of detecting the existence of biases in online argumentation. We first motivate the need for such algorithms, then define relevant terms and explain the relevant background. Drawing on this background, we then introduce three practical guidelines for any work into bias detection, and lay out a road map for the long-term goals of this project. Finally, we describe the primary contribution of this paper: an annotated dataset containing instances of personal attacks from real-world online arguments.

Biases in Online Argumentation

Biases have long been identified as problems plaguing fields involving expert reasoning, such as: legal reasoning (Chortek 2013), medicine and health (Campbell, Croskerry, and Petrie 2017; Daniel et al. 2017), forensic science (Lockhart and Satya-Murti 2017), and more. However the precise relation between biases and actual errors is not definitively established. Cognitive "de-biasing" strategies have mixed results (Sherbino et al. 2014; Lockhart and Satya-Murti 2017), and even medical professionals can have difficulty agreeing on the presence or absence of biases (Zwaan et al. 2017).

Nevertheless, while it may never be possible to completely eliminate biases, initial results suggest that there may be limited scenarios where awareness of common biases can decrease their harmful effects (Jenkins and Youngstrom 2016; Croskerry, Singhal, and Mamede 2013; Chew, Durning, and van Merriënboer 2016). For this reason, it is worth exploring whether the ability to detect potential biases in discussions and argumentation on internet forums and social media (hereafter simply referred to as "online argumentation") can demonstratively improve the quality of such discourse. For example, an online community may want to form a group that only allows high-quality comments that have been pre-screened for biases and fallacies, but lack the ability to appoint full-time, well-trained moderators whose job is to ensure such quality.

However, there are at least several major roadblocks to progress. First, there is some confusion and inconsistent usage of terms such as 'bias', 'fallacy', and so on. Perhaps due to these inconsistencies, popular understanding of fallacies tend to treat them either as the equivalent of checkmating in the game of argumentation, or as nuisances to be ignored

[1] http://www.stackexchange.com

[2] http://www.reddit.com/r/askscience

completely. Furthermore, those that wish to study biases or fallacies in online argumentation quantitatively find there is a severe shortage of datasets to do so. To start addressing this collection of problems, we will first define our terms.

Bias. We make use of several distinctions made by (Walton 1999)'s dialectical analysis of bias, which focuses on bias in argumentation, as opposed to focusing on the psychological states (e.g., personal beliefs, influences, etc.) of an arguer. According to his dialogical theory, a *bias* can simply be defined as a one-sided argument, which "advocates a particular proposition but fails to be balanced" (Walton 1999, p.79). One-sided arguments are contrasted with *two-sided arguments*, which fairly consider, weigh, and react to arguments and evidence on both sides of a dialogue.

For instance, an argumentative bias against a defendant in a murder trial might manifest as a willingness to accept weak evidence suggesting the defendant's guilt ("It's obvious that," "It's common sense to say," etc.), while at the same time remaining extremely critical or even outright ignoring evidence suggesting their innocence ("Of course he/she would say that," "That's just a coincidence," etc.). Later in this paper we will list Walton's (ibid) 9 indicators of bias.

Fallacies. Closely related to argumentative biases (and sometimes confused with them) are *formal and informal fallacies*. A fallacy is typically taken to be a particular argument type which is flawed in some way—due to a form that is not deductively valid (in the case of formal fallacies), or a simple non-sequitur (as with informal fallacies). Our treatment of fallacies will be in line with that of (Walton, Reed, and Macagno 2008)—Rather than treating the identification of fallacies in a dialogue as either winning moves or inconsequential nuisances, fallacies are properly classified as defeasible schemes which can be defeated by stronger schemes. In a way, such a treatment turns the game of argumentation into one of using the strongest possible scheme available to you, and attacking the weaknesses in an opponent's schemes, all of this happening in ways subject to the norms of the current dialogue.

In the definitions we use, a bias is something like a pattern or tendency of multiple arguments to skew in a certain direction, and a fallacy is a property of an individual argument. When we say that one's arguments are biased, it is a shortcutted way of saying that one's arguments show patterns (e.g., common fallacies) that can be explained by the existence of a bias.

Guidelines for Bias Detection

Even given the above definitions, there are still many other possible confusions about the causes, effects, and proper treatments of biases in argumentation. Therefore, before we present our proposed roadmap for developing a bias detector in online argumentation, it is important to make explicit a few guidelines, drawn from various sources. This non-exhaustive list of items partially formalizes our views on what exactly we are trying to capture when we are talking about bias in argumentation, and can serve as starting points for researchers to compare to their own views.

Biases are not necessarily 'bad', i.e., they do not always lead to fallacious reasoning or erroneous conclusions. The discovery of a bias in argumentation is properly treated as a reason to suspect that further arguments might also be biased, but they do not serve as knock-down, definitive determinations of fallacious arguments. Instead, their proper treatment is as *ad hominem* attacks; i.e., they should be treated as highly defeasible arguments which may be the strongest arguments available in some instances, but should be considered defeated in the presence of better arguments or evidence.

To further illustrate this point, consider that there are instances in which even *ad hominem* attacks can be considered appropriate. For a trivial example, imagine a Cretan who makes claim c: "No Cretans can make claims." Then we can simply put forward argument $\mathcal{A}$: by virtue of the fact that c is a claim, and the speaker is a Cretan, c is false. Because $\mathcal{A}$ hinges on the fact that the speaker is Cretan, it might be considered a form of *ad hominem* attack. But the existence of a fallacious form is insufficient to prove c is false. $\mathcal{A}$ can be easily re-formulated into a much stronger form with deductive validity if we consider the *a priori* truth that for any speaker who has any property, if that speaker makes an utterance which is false if the speaker does not have that property, then the speaker made a false utterance. $\mathcal{A}$ can thus be re-formulated into a deductively valid argument $\mathcal{A}'$, which will *defeat* any counterarguments to $\mathcal{A}$ (unless those counterarguments are as strong, or stronger, than $\mathcal{A}'$).

Just as the existence of a fallacy like *ad hominem* does not instantly disprove an argument's conclusion, the identification of a consistent bias affecting multiple arguments made by a single individual or organization does not necessarily disprove those arguments' conclusions. However, identification of a bias or fallacy in one's own arguments can be productive when it allows one to highlight potential weak points in their arguments, thus making it easier to search for counter-examples. It is for this reason that we believe that artificially intelligent software tools that can automatically detect biases in online argumentation may be able to improve the quality of discourse over what it is currently.

It may not be possible for reasoning to ever be completely free of biases. Another reason that biases should not be treated as instant disqualifiers is that they may simply be unavoidable. As (Walton 1999) observes of Jeremy Bentham's discussion of biases and fallacies, Bentham suggests at one point that "all our thinking and reasoning is biased, in the sense of being produced by an interest of some sort," namely a bias of interest. But "Bentham does not seem to think there is anything bad about this type of bias, or that the existence of it, in itself, is a sufficient reason to condemn reasoning as fallacious or faulty" (Walton 1999, p.16).

Walton's dialectical theory of bias (ibid) also notes that the context of the dialogue in which arguments appear must be considered, particularly its stated and unstated norms. A judge is expected to have minimal biases that are considered personal, but is also expected to act in a way that prefers certain values of justice, impartiality, nondiscrimination, and so on. The adherence to these values might be considered a type of bias, but one that is required by the norms of the relevant judicial systems. On the other hand, in a lighthearted debate between friends over their preferred sports teams, heavy partisanship on the part of the participants may actually be preferred. In fact, a reluctance to take sides and profess devotion to a sports team might earn one accusations of being a "fair-weather" or "bandwagon" fan.

Whenever possible, preserve information about how meaningful features contribute to determinations of bias. Current advances in deep learning have abetted the emergence of algorithms capable of performing complex classifications without using human-encoded mid-level features. Convolutional neural networks, for example, perform visual classification tasks without needing a human to teach the network how to recognize lines, edges, shapes, even artistic styles which can then be transferred to other images (Gatys, Ecker, and Bethge 2015).

However, such approaches do not necessarily learn features that, individually, are meaningful to human beings (though fixing this problem is an active area of research; see (Krause, Perer, and Ng 2016; Ribeiro, Singh, and Guestrin 2016)). There is a danger of creating bias detectors that may be able to reliably detect the possible existence of biases in online argumentation, without being able to explain *why* biases exist. The inability for such models to explain their conclusions in easily-understandable, non-esoteric terms can reduce their acceptance among the general public (Bornstein 2016). Furthermore, not being able to reliably identify contributing factors makes it difficult to produce actionable suggestions for how to reduce the negative effects of bias. For this reason, our approach is primarily bottom-up: we attempt to identify meaningful components and indicators of biases and create classifiers for those indicators first, before determining how they can be combined to create bias detectors.

A Roadmap for Detecting Problematic Reasoning

Taking into account the above principles for bias detection, particularly those which emphasize the importance of identifying recognizable components of bias, we suggest the following high-level strategy:

1. Identify meaningful indicators of bias, either of specific biases or biases in general. Such indicators, for pragmatic purposes, should also be detectable in online argumentation using current NLP technologies. We will list indicators of bias suggested by (Walton 1999) in the next section.
2. Create datasets for each indicator. After meaningful indicators of bias are chosen, datasets should be collected, ideally from real-world examples of online argumentation. Later in this paper, we will describe progress made in developing one such dataset.
3. Develop algorithms to reliably detect each indicator, or component features of indicators. The datasets collected in the previous step can be leveraged to create and test classification algorithms. It may be useful to create sub-classifiers designed to detect features that are believed to contribute to indicators, as we will do later in this paper. Such feature detectors can be combined later to produce indicator detectors, as is common in NLP work (Kiddon and Brun 2011; Wei and Wan 2017; Biyani, Tsioutsiouliklis, and Blackmer 2016). It is important, given the principles we have listed, that each indicator has its own classifier(s), even though multiple indicators will ultimately be used to determine the existence of bias.
4. Detect biases. It is a non-trivial step to determine precisely how the indicators can be leveraged to reliably detect the existence of biases. Presumably, much of this work would be performed in the first few steps—after all, indicators of bias shouldn't have been selected as indicators in the first place if they are not useful for detecting biases.

Indicators of Bias

Focusing only on argumentative bias then, is there a way to identify whether a set of arguments, presented in a text form typical of online argumentation, contains bias? There are at least two approaches that might allow us to answer this question. The first is to detect bias by using a kind of sentiment analysis on text or text fragments (Roman, Piwek, and Carvalho 2006; Roman et al. 2015). The second approach, which this paper favors, is to identify features that are normative indicators of bias, and to use these features as a starting point. In this section, we will list nine such indicators of bias in argumentation, originally identified by (Walton 1999, Ch. 4). Walton's approach is not empirical; rather, his "claim is a pragmatic and normative one, that these indicators are the ones that ought to be used as a [basis] for judging evaluations of bias in argumentation and are the ones that are practically most useful for this purpose" (ibid). His indicators are paraphrased below.

1. **Something to gain.** The arguer has something to gain from their argument's conclusion being true, or something to lose from being wrong. It should be noted that the use of this indicator is very reminiscent of the *ad hominem* fallacy. But as stated earlier, fallacies should be treated as defeasible argument schemes; i.e., if an arguer is accused of impartiality because they have something to gain or lose from the argument's correctness, and a stronger argument exists that can debunk that accusation, then the weaker argument is *defeated*. Otherwise, the accusation stands as an indicator of bias.
2. **Selectivity of arguments.** The arguer conveniently omits counterarguments to her own, or evidence harmful to their case.
3. **Lip-service selection.** Instead of ignoring counterarguments completely, they cite weak, easily dismissible, or misrepresented straw-man arguments that are easy to refute or make the other side look weak or non-credible in some way.
4. **Stated commitment to an identifiable position.** The arguer has an affiliation with, position with, or stated com-

mitment to, an organization or ideology that is relevant to the arguments being made. Like indicator (1), accusations of this indicator are highly defeasible and should not be treated as sufficient conditions for bias. Furthermore, Walton notes that it is implausible to expect speakers to be completely free of any beliefs, affiliations, or even biases whatsoever, and although “it would be unrealistic to expect anyone to be free of it [... it] can become a problem in argumentation when commitment is too firmly fixed to a position” (Walton 1999, p.98-99).

5. **Closure to opposed argumentation** A stubbornness or refusal to fairly consider arguments or evidence opposed to the speaker’s arguments. This indicator is best tested in interactive dialogues, because it shows up most prominently in an arguer’s responses to criticisms or strong counterarguments. Walton (ibid) also suggests this indicator is associated with the use of words and phrases designed to dissuade further thought or discussion, such as those identified by (Fearnside and Holther 1959): ‘it is obvious,’ ’everybody knows,’ ’every decent American wants,’ ’only a moron would believe,’ etc.
6. **Rigidity of stereotyping.** Stereotyping is another highly defeasible type of inference, in that it tends to produce weak arguments, but may be necessary in the extreme absence of further arguments or evidence. This indicator, like the previous, shows up in dialogues when faced with counterexamples to the stereotypes, an arguer refuses to show flexibility and acknowledge the defeat of the stereotype’s inference.
7. **Treating comparable cases differently.** This indicator effectively amounts to an arguer’s refusal to appreciate the similarity between two cases, where one case support’s the arguer’s views, whereas the second case does not. It is not clear, however, which objective standard (if any) for comparing two cases is to be used as a baseline for this indicator.
8. **Emphasis and hyperbole.** The arguer tends to use loaded words in a way that is imbalanced. For example, the arguer might describe testimonies in support of an opponent’s arguments as ’ridiculous’ or ’unconvincing,’ whereas testimonies on the other side might be ’bombshells’ or ’dramatic revelations’.
9. **Implicature and innuendo.** Closely related to the previous indicator, the arguer uses text that is worded in such a way, or otherwise suggests, “a conclusion or point of view that is highly argumentative and takes one side of a controversial issue” (ibid).

Given the roadmap to bias detection we laid out earlier, Walton’s indicators serve as a helpful starting point. However, not all of Walton’s indicators are equally applicable to online argumentation given the current state of natural language processing. As an initial attempt to create a dataset for these indicators, we have chosen to pursue the feature of *personal attacks*. Personal attacks are an important component of Walton’s indicators, contributing to many of them either directly or indirectly. For example, indicators (5) and (6) can manifest as personal attacks, such as insults (“only a moron would believe,” “you clearly don’t understand anything”), or refusal to consider opposing arguments due to the counter-arguer’s affiliation (“you’re just saying that because you’re a,” “of course you’d say that”). Therefore, the ability to reliably determine whether comments in online argumentation constitute personal attacks is useful.

FORM 1: “A says X, A has negative property P; therefore X is false”
FORM 2: “A says X, A has an interest in X being true; therefore X is false”
FORM 3: “A says X, A doesnt believe X or acts in a way Y which is inconsistent with X; therefore X is false”

Figure 1: The forms of argument considered to be personal attacks in our dataset.

There are at least three argument forms that are associated with personal attacks, and we define personal attacks in online argumentation as any comment which fits one of the forms listed in Figure 1.

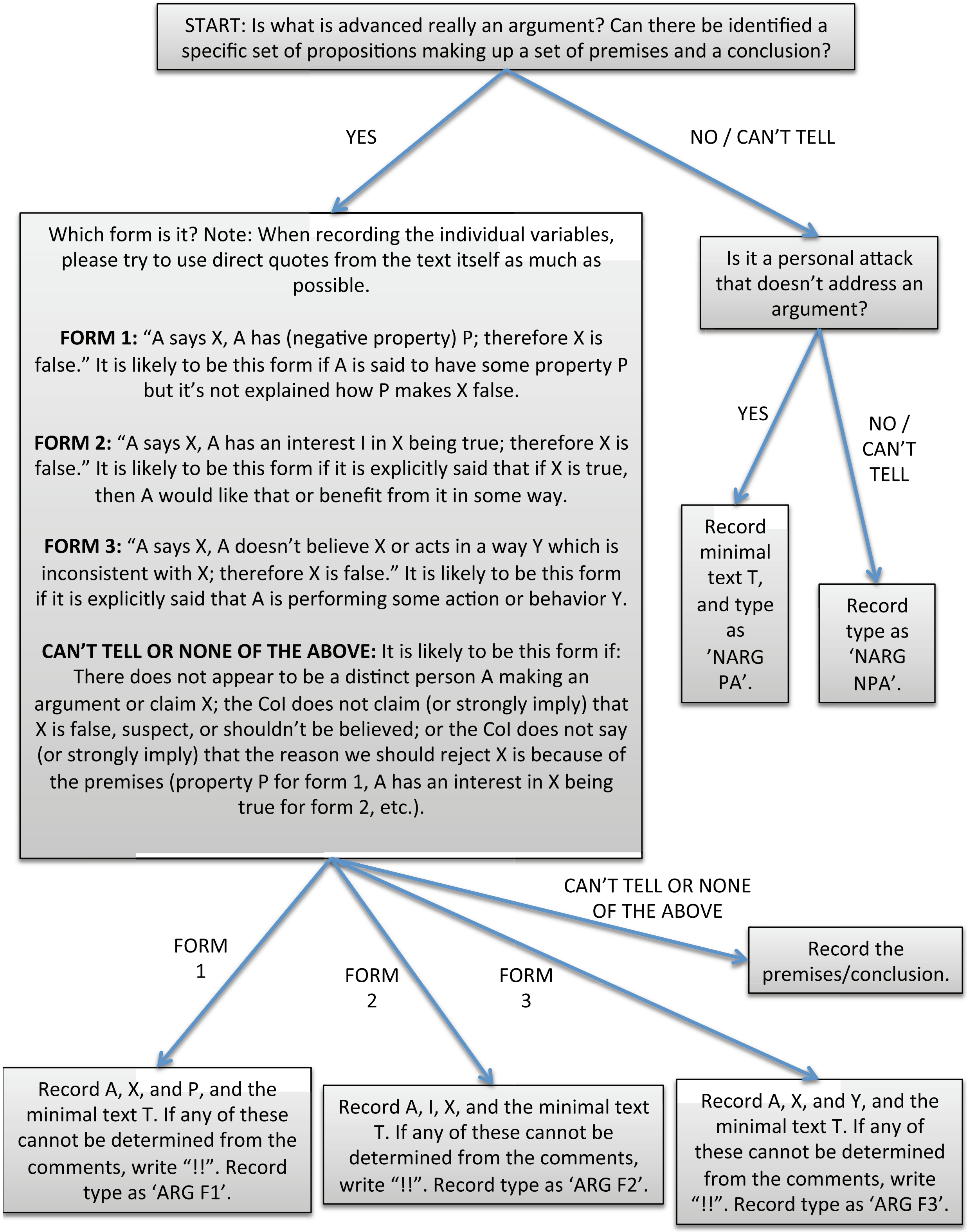
These forms are simplified versions of those identified by (Walton 1985). Walton’s analysis of the *ad hominem* fallacy also includes a set of critical questions for each of these forms, some of which we have borrowed for the creation of the dataset we will describe shortly.

An Algorithm for Detecting Personal Attacks

At present, we are not aware of any labeled datasets of personal attacks in Internet comments, and this absence hinders progress in the development of bias detectors. It is therefore worthwhile to build a dataset of personal attacks satisfying the conditions stated in our roadmap. However (and this will likely be the case for many bias indicators or components of indicators), acquiring labeled datasets is difficult. In part, this is due to the fact that objective and accurate assessments of the logical forms of arbitrary Internet comments requires a significant degree of training.

Perhaps more problematic, even properly trained individuals can disagree on the appropriate assessment for any given comment. Such disagreement is normally solved by creating a corpus of comments where multiple individuals assess each comment. Services such as Amazon Mechanical Turk are often useful for this purpose, and might be able to provide a sufficient amount of properly trained individuals. However, properly training and paying enough individuals to build a sizable dataset for each bias indicator can be extremely cost-prohibitive.

In order to address the problem, we have made some progress in developing a method for collecting and annotating comments containing personal attacks. This method is deployable to non-experts, since it can allow minimally trained individuals to annotate comments, and not cost-prohibitive, since it can be deployed to services like Amazon Mechanical Turk and not restricted to highly trained experts. This section will describe our progress, which at this stage is preliminary and in preparation for more rigorous validation.



START: Is what is advanced really an argument? Can there be identified a specific set of propositions making up a set of premises and a conclusion?

Which form is it? Note: When recording the individual variables, please try to use direct quotes from the text itself as much as possible.

FORM 1: "A says X, A has (negative property) P; therefore X is false." It is likely to be this form if A is said to have some property P but it's not explained how P makes X false.

FORM 2: "A says X, A has an interest I in X being true; therefore X is false." It is likely to be this form if it is explicitly said that if X is true, then A would like that or benefit from it in some way.

FORM 3: "A says X, A doesn't believe X or acts in a way Y which is inconsistent with X; therefore X is false." It is likely to be this form if it is explicitly said that A is performing some action or behavior Y.

CAN'T TELL OR NONE OF THE ABOVE: It is likely to be this form if: There does not appear to be a distinct person A making an argument or claim X; the CoI does not claim (or strongly imply) that X is false, suspect, or shouldn't be believed; or the CoI does not say (or strongly imply) that the reason we should reject X is because of the premises (property P for form 1, A has an interest in X being true for form 2, etc.).

Is it a personal attack that doesn't address an argument?

Record minimal text T, and type as 'NARG PA'.

Record type as 'NARG NPA'.

Record the premises/conclusion.

Record A, X, and P, and the minimal text T. If any of these cannot be determined from the comments, write "!!". Record type as 'ARG F1'.

Record A, I, X, and the minimal text T. If any of these cannot be determined from the comments, write "!!". Record type as 'ARG F2'.

Record A, X, and Y, and the minimal text T. If any of these cannot be determined from the comments, write "!!". Record type as 'ARG F3'.

Figure 2: The instructions provided to individuals in order to identify personal attacks in comments.

Step 1: Initial Filtering

Using the "Python Reddit API Wrapper" (PRAW),[3] we collected the 500 most recent submissions from each of the following subreddits[4]: politics, the_donald, askReddit, debateAnAtheist, debateAChristian, philosophy, news, worldnews, liberal, conservative, libertarian, Bad_Cop_No_Donut, Good_Cop_Free_Donut, gunpolitics, and gunsAreCool. From each submission, we collected the top comments, sorted by those that were most controversial (Reddit allows comments to be considered controversial if they contain a mix of upvotes and downvotes from other users). Altogether, we obtained 294,627 comments.

In order to initially filter through these comments, we looked for indicators that someone was making an argument that might be a personal attack, by searching for what we called "indicator comments"—i.e., comments that accuse others of committing some form of personal attack. For example, if a comment contains the words "ad hominem," then we have reason to suspect that it is an indicator comment which accuses its parent comment, the comment of interest (CoI), of committing an ad hominem, which is a type of personal attack. Although this technique returns many false positives (e.g., from a discussion on whether *ad hominem* is a fallacy), it gives us a way to identify enough test cases to initially validate our accessible identification algorithm (described shortly).

The full list of terms, in regular expression form, are: (F|f)ake news, a l(iar|ying), you('re|r) an*, (T|t)roll, (Y|y)ou('re|r) (so|just), (P|p)ersonal(ly)* attack, (A|a)d( )*hominem, post history, (O|o)f course (s)*he, and (O|o)f course you. Searching for these terms yielded 6,117 indicator comment/CoI pairs. We also retrieved the parent comment of the CoI, whenever one existed.

The terms used for the initial filtering step were selected by the authors' manually reading through Reddit discussions and identifying comments that seemed to follow personal attacks. The reasoning behind this approach is that our initial dataset can train weak classifiers that can then be used to create larger, more accurate labeled datasets of personal attacks. Not surprisingly, the initial filtering step showed some limitations of our chosen terms. For example, the term 'a l(iar|ying)' tended to be used by commenters who were performing, rather than calling out, personal attacks.

Some terms, such as 'fake news', appeared in comments that were clearly jokes, rather than comments making arguments or actual accusations of "fake news." With this revelation in place, we adjusted our personal attack identification algorithm to distinguish between arguments and non-arguments.

Step 2: An Accessible Algorithm for Identifying Personal Attacks

As stated earlier, it was important to develop an algorithm for identifying personal attacks in comments that is accessible to minimally-trained individuals, so that future creation of personal attack datasets would not be as cost-prohibitive as it currently is. We set about to classify comments as one of five types: three *ad hominem* attacks, arguments that are not of one of those three forms, non-arguments that are personal attacks, and non-arguments that are not personal attacks (the last of these categories was used as a default, catch-all category).

The three forms of *ad hominem* attacks recognized by our algorithm, pictured in Figure 1, correspond to three types of *ad hominem* fallacy, as described by (Walton 1985; van Heuveln).

Abusive Ad Hominem. An *ad hominem* attack is abusive if it contains an allegation of a property held by a speaker making a claim, and the conclusion that on the basis of the speaker having that property, the claim must be considered invalid. For example, a politician might be disbelieved because "he is a liar," or a news report might be considered false if the source is considered "fake news."

Circumstantial Ad Hominem. In the circumstantial *ad hominem*, a speaker making a claim is alleged to have an interest in the truth of the claim. An executive of a food company making a claim that his company produces the best food might be accused of only making that claim because of his position. Because the existence of a relevant interest distinguishes this form from the others, our algorithm requires that it be made explicit. Also note that the 'speaker' in this fallacy (as with the other two forms) can be an individual, an organization making an official statement, and so on.

Inconsistency Ad Hominem. Although this type of *ad hominem* is not as frequently recognized as the others (some lump it in with the circumstantial variety), it is helpful to distinguish. An *ad hominem* is inconsistent if it involves a speaker who makes claim C, but the speaker also performs some particular action, has some belief, or makes other claims that are inconsistent with C. A parent telling her child that "smoking is bad" when she is a heavy smoker herself might be attacked with an inconsistency *ad hominem*. In general, accusations of hypocrisy will satisfy the conditions of this form.

Considering the above, we developed the algorithm pictured in Figure 2, whose instructions were arrived at over several iterations of trial and error.

Verifying the Dataset

The wording shown in Figure 2 was iteratively modified in order to reduce possible sources of confusion for minimally-trained individuals. To measure its success towards this goal, four undergraduate students who did not have any formal training in formal logic were asked to use the algorithm in

[3] https://github.com/praw-dev/praw

[4] To access any of these subreddits, simply append the subreddit name to 'http://reddit.com/r/'. For example, the URL for the politics subreddit is: http://reddit.com/r/politics.

Figure 2 on the same set of comments. 100 comments were randomly selected from the set of initially filtered comments described earlier, and presented individually to each student in the same order. They were instructed not to communicate with each other, to avoid having their answers influence each other.

To measure how well the four students agreed with each other, we use Krippendorff's alpha-reliability measure α (Krippendorff 2004).[5] α attempts to measure how well multiple raters agree when asked to rate the same list of items:

$$\alpha = \frac{A_o - A_e}{1 - A_e}$$

where A_o is the percent of all observed matches within units and A_e is the percent of matches obtainable by chance (Krippendorff 2004). Furthermore, for some possibly observable value c, o_{cc} is the total number of times that on some unit to observe, a rater agreed with another that the correct observation was c, and n_c is the number of times that c was observed overall. If n is the total number of observations:

$$A_o = \frac{\sum_c o_{cc}}{n} \qquad A_e = \frac{\sum_c n_c(n_c - 1)}{n(n-1)}$$

An α value closer to 0 means there was an agreement between raters no better than chance, and an α closer to 1 indicates more agreement. Three such calculations were made:

Simple condition. For each comment, each student would determine exactly one of the six possible classifications as described in Figure 2. The simple condition calculated α for these classifications with no further preprocessing.

High confidence condition. For each comment where a classification was made, the student was asked to rate their confidence in their assessment on a three-point scale. In the high confidence condition, only assessments with the highest confidence rating (3) were taken into account; all other assessments were treated as missing data.

Text agreement condition. Because we evaluated unfiltered comments on Reddit forums, there was enormous variation in the length and number of arguments contained within each comment. Some contained only a few words, others contained multiple paragraphs. However, our student raters were only asked to choose one argument per comment to rate. This led to the situation where students who chose different arguments within the same comment to rate were treated as disagreeing with each other.

The text agreement conditions attempted to rectify this problem by requiring all students, when classifying comments, to record the "minimal text" of the argument; i.e., to directly copy and paste the portion of the comment that contains all of the text necessary for the argument being considered, with as much unnecessary text removed as possible.

Condition	α	**n**
Simple	0.3946	309
High confidence	0.4293	186
Text agreement	0.6033	178
Text agreement + confidence	0.4961	114

Table 1: The alpha values of the four conditions we tested.

For example, a comment that says "You're an idiot, so I don't believe you. I'm going to delete my account." would have the minimal text "You're an idiot, so I don't believe you" or "You're an idiot", depending on which type of argument form was identified.

In order to approximately measure whether two students identified the same argument fragment to focus on, we used the string distance measure implemented by Python's difflib.SequenceMatcher algorithm (Ratcliff and Metzener 1988). Given any two ratings for the same item, if the minimal text has a similarity score of 0.6 or higher, then it is considered by the text agreement condition. In the text agreement + confidence condition, we additionally only consider a pair of ratings if they also both have confidence values of 3. The resulting values, along with the number of values considered n, are listed in Table 1.

Conclusion and Next Steps

The goal of the present paper was two-fold. First, we introduced a high-level plan for research into bias detection in online argumentation, by describing guidelines that can be followed in such research and then laying out a series of steps to carry out. The second step in our proposed steps is to create datasets for each indicator of bias in argumentation, and the second goal of this paper was to do just that. Settling on indicators related to personal attacks, we created an initial dataset which we believe is in line with the high-level plan introduced here.

We hope that the principles and roadmap introduced here will encourage discussion and motivate further research into detecting biases in argumentation. Of course, there is much more work to do than can be described in this paper. For example, there may be further possible refinements to the flowchart algorithm in Figure 2 which can be performed. The effects of such changes can be quantified by comparing their resulting α values to those reported in Table 1. We hope to raise α to a level where we can be confident enough to offer our algorithm to completely untrained users on Amazon Mechanical Turk. This would allow us to create a substantially larger labeled corpus of comments, which may be used to train at least weak classifiers; these can then be used to search the set of comments we collected before the initial filtering step to find personal attacks not captured by the terms we manually identified.

Acknowledgements

This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-17-1-0191. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the au-

[5] Our implementation was specifically based on the variation designed for nominal data, an unlimited number of participants/observers, and possibly missing data described at: http://web.asc.upenn.edu/usr/krippendorff/mwebreliability5.pdf.

thors and do not necessarily reflect the views of the United States Air Force.

References

Biyani, P.; Tsioutsiouliklis, K.; and Blackmer, J. 2016. " 8 amazing secrets for getting more clicks": Detecting clickbaits in news streams using article informality. In *AAAI*, 94–100.

Bornstein, A. M. 2016. Is Artificial Intelligence Permanently Inscrutable? *Nautilus* 40.

Campbell, S. G.; Croskerry, P.; and Petrie, D. A. 2017. Cognitive bias in health leaders. In *Healthcare Management Forum*, volume 30, 257–261. SAGE Publications Sage CA: Los Angeles, CA.

Cheng, J.; Bernstein, M.; Danescu-Niculescu-Mizil, C.; and Leskovec, J. 2017. Anyone can become a troll: Causes of trolling behavior in online discussions. *arXiv preprint arXiv:1702.01119*.

Chew, K. S.; Durning, S. J.; and van Merriënboer, J. J. 2016. Teaching metacognition in clinical decision-making using a novel mnemonic checklist: an exploratory study. *Singapore medical journal* 57(12):694.

Chortek, M. 2013. The psychology of unknowing: Inadmissible evidence in jury and bench trials. *The Review of Litigation* 32(117).

Croskerry, P.; Singhal, G.; and Mamede, S. 2013. Cognitive debiasing 1: origins of bias and theory of debiasing. *BMJ Qual Saf* bmjqs–2012.

Daniel, M.; Khandelwal, S.; Santen, S. A.; Malone, M.; and Croskerry, P. 2017. Cognitive debiasing strategies for the emergency department. *AEM Education and Training* 1(1):41–42.

Fearnside, W. W., and Holther, W. B. 1959. *Fallacy: The Counterfeit of Argument*. Englewood Cliffs, NJ: Prentice Hall.

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2015. A Neural Algorithm of Artistic Style. *CoRR* abs/1508.06576.

Jenkins, M. M., and Youngstrom, E. A. 2016. A randomized controlled trial of cognitive debiasing improves assessment and treatment selection for pediatric bipolar disorder. *Journal of consulting and clinical psychology* 84(4):323.

Kiddon, C., and Brun, Y. 2011. That's what she said: Double entendre identification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, 89–94. Association for Computational Linguistics.

Krause, J.; Perer, A.; and Ng, K. 2016. Interacting with Predictions: Visual Inspection of Black-box Machine Learning Models. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, 5686–5697. New York, New York, USA: ACM.

Krippendorff, K. 2004. *Content Analysis: An Introduction to Its Methodology*. Thousand Oaks, CA: Sage Publications, Inc., 2 edition.

Lockhart, J. J., and Satya-Murti, S. 2017. Diagnosing crime and diagnosing disease: bias reduction strategies in the forensic and clinical sciences. *Journal of forensic sciences*.

Ratcliff, J. W., and Metzener, D. E. 1988. Pattern matching: The gestalt approach. *Dr Dobbs Journal* 13(7):46.

Ribeiro, M. T.; Singh, S.; and Guestrin, C. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *CoRR* abs/1602.04938.

Roman, N. T.; Piwek, P.; Carvalho, A. M. B. R.; and Alvares, A. R. 2015. Sentiment and Behavior Annotation in a Corpus of Dialogue Summaries. *Journal of Universal Computer Science* 21(4):561–586.

Roman, N. T.; Piwek, P.; and Carvalho, A. M. B. R. 2006. Politeness and bias in dialogue summarization: Two exploratory studies. In Shanahan, J. G.; Qu, Y.; and Wiebe, J., eds., *Computing Attitude and Affect in Text: Theory and Applications*. Springer Netherlands. 171–185.

Sherbino, J.; Kulasegaram, K.; Howey, E.; and Norman, G. 2014. Ineffectiveness of cognitive forcing strategies to reduce biases in diagnostic reasoning: a controlled trial. *Canadian Journal of Emergency Medicine* 16(1):34–40.

van Heuveln, B. The ad hominem fallacy - critical thinking.

Walton, D.; Reed, C.; and Macagno, F. 2008. *Argumentation Schemes*. Cambridge University Press.

Walton, D. 1985. *Arguer's Position: A Pragmatic Study of Ad Hominem Attack, Criticism, Refutation, and Fallacy*. Greenwood Press.

Walton, D. 1999. *One-Sided Arguments: A Dialectical Analysis of Bias*. State University of New York Press.

Wei, W., and Wan, X. 2017. Learning to identify ambiguous and misleading news headlines. *arXiv preprint arXiv:1705.06031*.

Woolley, S. C., and Howard, P. N. 2016. Automation, algorithms, and politics— political communication, computational propaganda, and autonomous agents—introduction. *International Journal of Communication* 10:9.

Zwaan, L.; Monteiro, S.; Sherbino, J.; Ilgen, J.; Howey, B.; and Norman, G. 2017. Is bias in the eye of the beholder? a vignette study to assess recognition of cognitive biases in clinical case workups. *BMJ Qual Saf* 26(2):104–110.

A Personalized Method for Calorie Consumption Assessment

Yunshi Liu

College of Information Science and Engineering, Ritsumeikan University, Shiga, Japan
is0388ik@ed.ritsumei.ac.jp

Pujana Paliyawan, Takahiro Kusano

Graduate School of Information Science and Engineering, Ritsumeikan University, Shiga, Japan
pujana.p@gmail.com, is0212kf@ed.ritsumei.ac.jp

Tomohiro Harada, Ruck Thawonmas

College of Information Science and Engineering, Ritsumeikan University, Shiga, Japan
{harada@ci, ruck@is}.ritsumei.ac.jp

Abstract

This paper proposes an image-processing-based method for personalization of calorie consumption assessment during exercising. An experiment is carried out where several actions are required in an exercise called broadcast gymnastics, especially popular in Japan and China. We use Kinect, which captures body actions by separating the body into joints and segments that contain them, to monitor body movements to test the velocity of each body joint and capture the subject's image for calculating the mass of each body joint that differs for each subject. By a kinetic energy formula, we obtain the kinetic energy of each body joint, and calories consumed during exercise are calculated in this process. We evaluate the performance of our method by benchmarking it to Fitbit, a smart watch well-known for health monitoring during exercise. The experimental results in this paper show that our method outperforms a state-of-the-art calorie assessment method, which we base on and improve, in terms of the error rate from Fitbit's ground-truth values.

Introduction

It is suggested by several health experts that people should be concerned of their calorie intake and consumption (Hill et al. 2003). Nowadays, the assessment of calorie consumption remains challenging. There exists a gas analysis system for calorie consumption assessment (B Böhm, Hartmann, and H Böhm 2016), which seems highly accurate, but it needs large space and expensive devices. In addition, users of their system also lose freedom to move. Another method (Tsou and Wu 2015) was developed by Tsou and Wu where Kinect, a line of motion sensing input device that can detect the gesture of a whole body, is used for calorie assessment. This kind of device is expected to be extensively used in constructing rehabilitation applications in calorie assessment that are related to health promotion (Da Gama et al. 2015). In Tsou and Wu's method, the coordinates of body joints in 3D space are captured by Kinect and used to calculate the velocity of each joint movement, and then a kinetic energy for estimating calorie consumption. The method yields promising performance; however, there are still issues that can be improved, in particular, the issue that assessment does not take the body size of individual users into account.

In this paper, we propose an improved version of the method by Tsou and Wu. Note that in their method, kinetic energies are computed by using the velocities of body joints and the *standard* mass of each joint (a mass represents the portion of a joint of interest to the whole body, including muscles and bones attached to that joint). On the contrary, in our work, the mass of each body joint is derived by processing an image of the subject's body. In other words, calorie consumption assessment by our method takes the body size of each user into account. Following an existing protocol for system evaluation (Ryu, Kawahawa and Asami 2008), we use a reference device, Fitbit, to evaluate the assessment accuracy of our system.

Existing Work

Nowadays, we could know how many calories a human consumes during walking by some smartphone applications. But accuracy is still in question. Most applications do not consider mass, which means they do not weight the importance of each body segment. Therefore, a method is required that adapts to the body size and weight of each individual user**.**

For the aforementioned exiting work on calorie consumption assessment based on gas analyzing, we stated that, based on their result, it is an accurate system. However, considering a high cost, largely needed space, it is impractical to adopt their approach to applications for promoting users' physical health through daily exercise or motion gaming.

Our work is mainly based on the aforementioned existing method by Tsou and Wu, in which Kinect is used to monitor users' activities and assess their calorie consumption. They showed error rates to a ground truth that is calorie consumption assessed by a reliable assessment tool, i.e., a heart rate monitor. In addition, the longer the training time, the less the error rate. They used kinetic energies of the body joints to build a regression function for estimating calorie consumption. The kinetic energy of each body joint is calculated as a multiplication of the joint's standard scale with the body weight. We conjecture that assessment can be improved if the body scale is measured specifically for each individual user.

Methodology

According to Tsou and Wu's method, kinetic energy parameters are used to assess calorie consumption. This shows that such energies are related to the calorie consumption amount. Following their recipe, we also use kinetic energy parameters to assess calorie consumption.
The kinetic energy needs mass and velocity to calculate. In Tsou and Wu's method, the kinetic energy in each joint is used in multiple linear regressions for predicting calorie consumption. The assessment of mass, velocity, and calorie consumption are described in the subsections below, respectively.

Mass

Tsou and Wu's method assumes that the shape of body is universal to all people while in our method, the system obtains mass by analyzing the body shape of each user specifically. Image processing is done on a depth image (An example is shown in Fig.1), where the ratio of each body segment to the whole body is computed and used to represent the mass percentage of each joint. By multiplying the mass percentage with the weight of the user, we obtain the mass of each part for calculation of the energy. To obtain the mass for each of Kinect's 20 joints, we used software called *ImageJ* to measure the ratio of the number of pixels in each joint's area to that in the whole body.

Velocity

While a user is exercising, the system obtains his/her streaming skeleton data from Kinect (see *Figure 1*). The skeleton data represent 3D coordinates of all body joints in each row. We set the data frame rate to 25 fps. We derive the velocity of a given joint over a period of time by using the differentiation method.

The differentiation method is widely used in physics to obtain the average velocity over time. When the period is very short, we can regard this average velocity as the instantaneous velocity, i.e., the formula of which at time *t* for joint *j* is as follows:

$$v_{j,t} = \frac{ds}{dt} \tag{1}$$

where *ds* is the distance that joint *j* moves during the interval [*t* - *dt*, *t*]. For each joint, all instantaneous velocities are collected for the assessment of its kinetic energy.

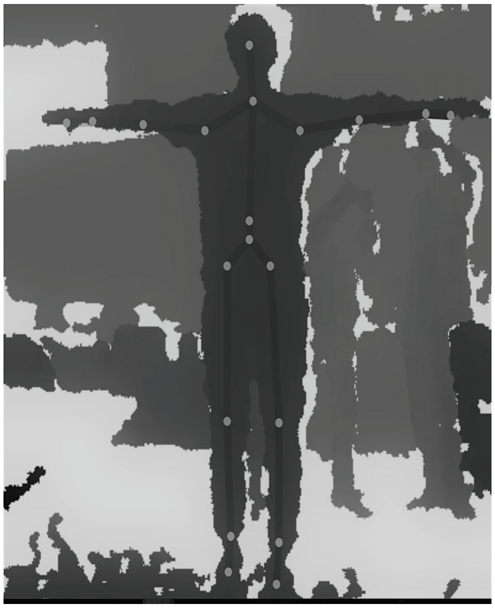


Figure 1 An Example of a Depth Image

Calorie Consumption

After obtaining the mass and velocity data of all body joints, we compute the kinetic energy for each one. As done in Tsou&Wu's original method, the values from three dimensions are used to calcuate the kinetic energy E_j for joint j as follows:

$$E_j = \left|E_{j,x} + E_{j,y} + E_{j,z}\right| \tag{2}$$

In Eq. (2), the parameters $E_{j,x}, E_{j,y}, E_{j,z}$ represent the kinetic energy in each dimension. Classical mechanics indicate that the kinetic energy E of a particle of mass M travelling at speed V is given by $E = 1/2MV^2$. As a result, Eq. (2) can be reformulated as follows:

$$E_j = \left|\frac{1}{2}M_jV_{j,x}^2 + \frac{1}{2}M_jV_{j,y}^2 + \frac{1}{2}M_jV_{j,z}^2\right| \tag{3}$$

In actual calculation, the velocity for each dimension in Eq. 3 is combined as one velocity. The parameter M_j in this equation represents the mass of body joint j. This mass is obtained by Eq. 4 where a_j is the ratio of joint j in comparison to the area of the whole body, as described in Subsection **Mass**, and *weight* is the weight of a user of interest.

$$M_j = weight \times a_j \tag{4}$$

Note that E_j in Eq. 2 is the kinetic energy at a given short period, e.g., 1 second. By accumulating this amount over the whole exercise session of, say, T seconds, we obtain K_j as an accumulated energy, or in other words the total energy spent for an exercise of interest (Eq. 5).

$$K_j = \sum_{t=1}^{T} E_{j,t}, \tag{5}$$

Table 1 Twenty Joints in Kinect

Head	Center Shoulder	Left Shoulder	Right Shoulder
Left Elbow	Right Elbow	Left Wrist	Right Wrist
Left Hand	Right Hand	Spine	Center Hip
Left Hip	Right Hip	Left Knee	Right Knee
Left Ankle	Right Ankle	Left Foot	Right Foot

where $E_{j,t}$ is

$$E_{j,t} = \frac{1}{2}M_jv_{j,t}^2 \tag{6}$$

Eq. 5 is applied to each of the 20 body parts (see *Figure 2*). Following the recipe in Tsou and Wu's method, calorie consumption (*CC*) is computed by using a multiple regression function having the resulting energies as input (Eq.7 where $b_0 \sim b_{20}$ indicate a bias and the coefficient for each dimension, respectively). The regression function is constructed in a training stage, in which *CC* from Fitbit is used as the dependent variable and the energies of all body parts are used as the independent variables in an analysis to find $b_0 \sim b_{20}$.

$$CC = b_0 + \sum_{j=1}^{20} b_jK_j \tag{7}$$

Eq. 7 is used for calculation of calorie consumption in our experiment for both Tsou and Wu's method and our method. For the former, a_j in Eq. 4 is set to a standard scale of the human body. However, since in their work, some joints are combined, we need to separate them in order to have 20 joints as in our method. By checking the joints that didn't appear in their method, we found that the "body" part (30 percent of the whole body) mentioned in their method contains five parts of joint: Center Shoulder, Left and Right Shoulder, Spine, Center Hip and Left and Right Hip. As how each part contributes to Tsou and Wu's body remains unknown, we simply considered that all segments related to those joints share the same mass percentage, and when considering symmetric parts, we future divided the percentage into half. The 20 segments, each corresponding to one of Kinect's 20 joints (Table 1), and the standard scale for any subject are shown in Table 2.

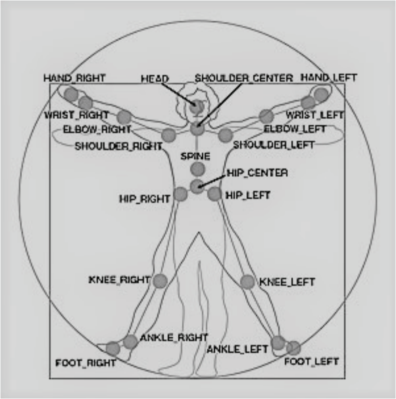


Figure 2: The Concept Map of Joints in Kinect (NikkeiBP 2012)

Table 2 Standard Scale for Any Subject

Segment Name	Percentage	Accumu-lated %.	Joint Number
Head	10%	10%	1
Left, Right Elbow	4%*2=8%	18%	2,3
Left, Right Wrist	3%*2=6%	24%	4,5
Left, Right Hand	2.5%*2=5%	29%	6,7
Center Shoulder	6%	35%	8
Left, Right Shoulder	3%*2=6%	41%	9,10
Spine	6%	47%	11
Center Hip	6%	53%	12
Left, Right Hip	3%*2=6%	59%	13,14
Left, Right Knee	10%*2=20%	79%	15,16
Left, Right Ankle	7%*2=14%	93%	17,18
Left, Right Foot	3.5%*2=7%	100%	19,20

Experiment

We evaluated our system on two different sets of motions from broadcast gymnastics (BG): one by NHK (JPN[1]), Japan's national public broadcasting organization, and another by by Chinese Sports Government (CHN[2]). They are exercises that are popular and widely known among people in each respective nation. As a result, we used these sets of motions in our experiment.

Process

For six subjects, each will be asked to do either JPN or CHN, depending on their choice, for construction of the prediction model. Figure 3 shows a subject doing an exercise in the experiment. This takes approximately 30 minutes. There are three steps as follows.

First, according to the method provided by Taylor (Taylor et al. 2012), before or after an experiment, a subject wears Fitbit and rests for 5 minutes. During such a period, the calorie consumption result from Fitbit is acquired. This data is required to ensure the measurement goes well by verifying whether the value in resting is not higher than the value in exercising.

Second, the subject is asked to do JPN or CHN, either on their own after given a guidance or following an exercise video, while wearing Fitbit. Then after finishing exercise, the calorie consumption data from Fitbit are collected, and the first and second steps will be repeated twice.

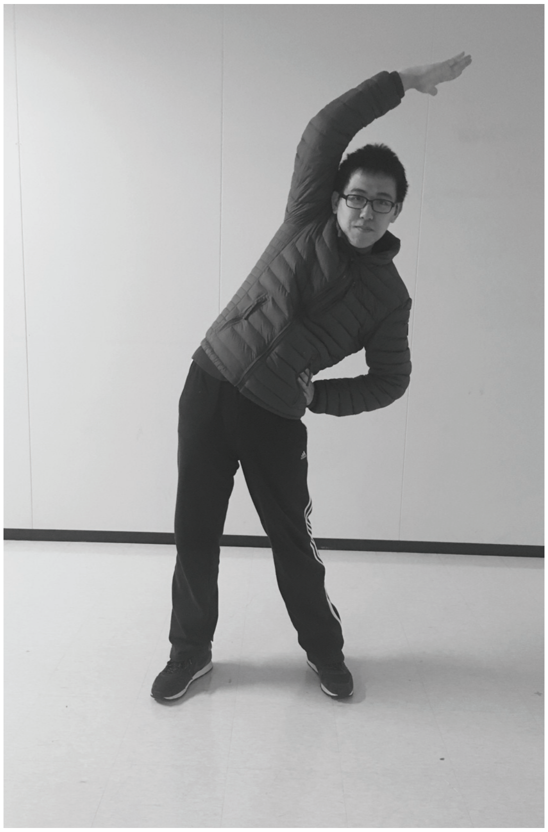


Figure 3: A subject performing a broadcast gymnastics

Third, when the three cycles for the first and second steps are finished, a photo of the subject is taken with his/her hands stretched up. This photo contains 20 joints. It is used in image processing to obtain the mass scale of body joints.

Data

There are two types of data in our experiment: the ground truth data from Fitbit and the mass data.

The ground truth is the calorie consumption assessed by Fitbit, both during the resting and exercising (engaging in JPN or CHN) time of the experiment. In this experiment, there are six subjects (three subjects from Japan, three subjects from China) for evaluating the prediction model of calorie consumption. Each subject did BG three times. The data are shown in Table 3, where *i*R represents the calorie loss the rest time before the *i*st exercise, and *i*E represents the calorie loss in the *i*st exercise. From this set of data, it can be seen that the calorie consumption in the rest situation (marked as 1R, 2R, 3R) is not more than the consumption in exercising (marked as 1E, 2E, 3E). The results from 1E, 2E, and 3E are used in comparison of the two prediction models.

[1] JPN Broadcast Gymnastics, 1st version, https://www.youtube.com/watch?v=b4SH_lap4ag

[2] CHN 9th National Broadcast Gymnastics official, http://www.iqiyi.com/w_19rqvi3qt9.html

Table 3 Ground Truth Data from Fitbit(Data Unit: kcal)

Name	1R	1E	2R	2E	3R	3E
Sub.1	12	16	17	20	16	19
Sub.2	9	25	9	26	7	29
Sub.3	6	14	7	19	6	15
Sub.4	7	20	9	17	8	19
Sub.5	9	22	14	26	6	21
Sub.6	11	31	13	30	26	33

Table 4 Example of Mass Data of a Subject

Segment Name	Percentage	Total
Head	5.76%	5.76%
Center Shoulder	9.99%	15.75%
Left Shoulder	5.49%	21.24%
Right Shoulder	5.49%	26.73%
Left Elbow	3.07%	29.80%
Right Elbow	3.07%	32.87%
Left Wrist	1.40%	34.27%
Right Wrist	1.40%	35.67%
Left Hand	1.05%	36.72%
Right Hand	1.05%	37.77%
Spine	10.47%	48.24%
Center Hip	3.64%	51.88%
Left Hip	4.36%	56.24%
Right Hip	4.36%	60.60%
Left Knee	9.64%	70.24%
Right Knee	9.64%	79.88%
Left Ankle	7.10%	86.98%
Right Ankle	7.10%	94.08%
Left Foot	2.96%	97.04%
Right Foot	2.96%	100.00%

The mass data shows the body scale of each subject. It is unique to each subject as shown for example in Table 4. This data is multiplied by the subject's weight for each segment (Eq. 4) in order to obtain the segment's mass.

Performance Metric:

The metric for performance evaluation is shown in Eq. 8. This metric shows the error rate in calorie consumption assessment for the n^{th} subject in the i^{th} exercise where E_{fn}^{i} is the result from Fitbit and E_{kn}^{i} is the result from a prediction model of interest.

$$Error\_rate(n,i) = \frac{\left|E_{fn}^{i} - E_{kn}^{i}\right|}{E_{fn}^{i}} \qquad (8)$$

Results and Analysis

The results are shown in three parts: error results, cross validation results, and statistical test results. Error results are the evaluation results over training data, which indicate that our method outperforms Tsou and Wu's method. Crossvalidation results ensure that the proposed should work well even on unseen data. Statistical test results indicate that there is a statistically significant difference between the two methods in cross validation.

Error Results

We compared CC measured by Tsou and Wu's and by our method to the ground truth provided by Fitbit. We benchmarked the two methods using the error rate (Eq. 8). Table 5 shows that our method yields less error rate than Tsou and Wu's method. In addition, the error rate of our method is obviously smaller than Tsou and Wu's method, which means we have successfully improved state-of-the art Tsou and Wu's method in predicting calorie consumption.

Table 5 Error Rates of Our Method and

Tsou & Wu's Method over the Training Data

Subject	Ours	Tsou & Wu's
1	$1.39*10^{-5}$	$1.79*10^{-5}$
2	$4.35*10^{-5}$	$1.93*10^{-2}$
3	$1.86*10^{-5}$	$3.23*10^{-5}$
4	$1.54*10^{-5}$	$3.58*10^{-5}$
5	$1.54*10^{-5}$	$2.14*10^{-5}$
6	$3.18*10^{-5}$	$4.32*10^{-5}$

Cross Validation Results

In order to confirm the accuracy of the prediction model, we ran a 3-fold cross validation. In this cross validation, part of the data of all subjects (e.g., data from the first and second exercises for all subjects) are used for constructing a prediction model for each method, then the prediction models are tested on the remaining data (e.g., referring to the example above, data from the third exercise); this is done three times, each with a different combination of training and testing data, for each method in order to obtain the average result.

Table 6 shows the error rates in cross validation. Note that the error rate of our method (Tsou and Wu's) at the i^{st} exercise, Ours_i (Tsou & Wu_i), shows the performance of the prediction model based on the corresponding method using the data in the remaining exercises for training. As can be seen from the table, the error rates of the proposed method are in most cases less than Tsou and Wu's method.

Statistical Test Results

We conducted a Wilcoxon signed-rank test to find whether there is a statistically significant difference between error rates from the two methods. The resulting *p* value is 0. 00854, which is less than 0.01, indicating that there is a significant difference at the confident interval of 99%. As a result, we can state that our attempt to improve Tsou and Wu's method through personalization of the user's mass scale is successful.

Conclusions and Future Work

We have presented a personalized method for calorie consumption assessment using Kinect based on the unique shape of each user. Kinect can produce skeleton data for analyzing the movements of body joints that lead to the velocity of each joint, and depth images that lead to mass data that are unique to each subject, both of which enable kinetic energy calculation. We build a prediction model based on the results from the ground truth data that connects the kinetic energies from Kinect. By comparing to the prediction model by Tsou and Wu, which uses standard scale mass data on every subject, our method utilizing personalized mass data outperforms Tsou and Wu's method, both in evaluation over training data and in evaluation using cross validation.

In future work, we will employ this method to monitoring the health state of motion-game players. This can be done by constructing a calorie consumption system that uses Kinect and a ground truth device for a prediction model, and by considering the effect of the amount of exercises (Slentz et al. 2004). We will also add a potential energy into the assessment formulas and estimate post-exercise calorie burn. In addition, our method can be used for health monitoring during full-body motion gaming to promote a healthy exercise while preventing injuries.

Table 6 Error Rates of Our Method and Tsou & Wu's Method for Each Testing Data in a Three-Fold Cross Validation

Subject	Ours$_1$	Ours$_2$	Ours$_3$	Tsou & Wu's$_1$	Tsou & Wu's$_2$	Tsou & Wu's$_3$
1	0.2106	**0.2196**	0.1408	**0.1537**	0.1464	**0.1266**
2	**0.0303**	**0.0208**	**0.1429**	0.3009	0.5538	0.3447
3	**0.1166**	**0.0834**	0.3677	0.3409	0.5433	**0.4870**
4	0.3960	**0.1634**	**0.0501**	**0.2506**	0.3803	0.3096
5	**0.1045**	**0.0458**	**0.1501**	0.2227	0.2724	0.1927
6	**0.0529**	0.6126	0.3941	0.2040	**0.5131**	**0.4292**
Avg.	**0.1835**			0.3207		

Acknowledgement

The authors wish to thank members of Intelligent Computer Entertainment Lab, Ritsumeikan University, for their helps in the experiment and suggestions.

References

Böhm, B., Hartmann, M., and Böhm, H. 2016. Body segment kinematics and energy expenditure in active video games. *Games for health journal* 5.3 (2016): 189-196.

Da Gama, A., Fallavollita, P., Teichrieb, V. and Navab, N. 2015. Motor rehabilitation using Kinect: A systematic review. *Games for health journal*, 4(2): 123-135.

Hill, J. O., Wyatt, H. R., Reed, G. W. and Peters, J.C. 2003. Obesity and the environment: where do we go from here? *Science* 299.5608 (2003): 853-855.

NikkeiBP, The Concept Map of Joints in Kinect, 2012. http://itpro.nikkeibp.co.jp/article/COLUMN/20120410/390410/

Paliyawan, P., Kusano, T., Nakagawa, T., Harada T. and Ruck T. 2017. Adaptive Motion Gaming AI for Health Promotion. *AAAI Spring Symposium Series (AAAI-17),* 720-725.

Ryu, N., Kawahawa, Y., and Asami, T. 2008. A calorie count application for a mobile phone based on METS value. In *Sensor, Mesh and Ad Hoc Communications and Networks, SECON'08. 5th Annual IEEE Communications Society Conference on*, 583-584. IEEE.

Slentz, C. A., Duscha, B. D., Johnson, J. L., Ketchum, K., Aiken, L. B., Samsa, G. P., ... and Kraus, W. E. 2004. Effects of the amount of exercise on body weight, body composition, and measures of central obesity: STRRIDE—a randomized controlled study. *Archives of internal medicine* 164(1): 31-39.

Taylor, L. M., Maddison, R., Pfaeffli, L. A., Rawstorn, J. C., Gant, N., and Kerse, N. M. 2012. Activity and energy expenditure in older people playing active video games. *Archives of physical medicine and rehabilitation* 93(12): 2281-2286.

Tsou, P. F., and Wu C. C. 2015. Estimation of calories consumption for aerobics using Kinect based skeleton tracking. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, 1221-1226. IEEE.

IoT-Based Emotion Recognition Robot to Enhance Sense of Community in Nursing Home

Shintaro Nagama, Masayuki Numao
Department of Communication Engineering and Informatics
The University of Electro-Communications
n1411227@edu.cc.uec.ac.jp

Abstract

Senior isolation is becoming a major social problem in Japan, as a super-aged society where more than a quarter of population is over 65 years old. Many elderly people are living in single-resident homes without family or social support. Even in nursing home, residents stay in their private bedrooms lonely without participating social activities, such as chatting, playing game, watching TV together at a living room, etc.

Since social isolation leads to serious consequences such as disuse syndrome, mental depression, suicide etc., maintaining person's sense of community is very important. But measuring sense of community is difficult because it is a mental process and many kinds of activities and interactions are involved in the process. In this paper, we define Social Activities of Daily Living (SADL) to focus on social activities to enhance the sense of community. We also propose a multimodal sensor based recognition method for SADL, which is implemented in the IoT-based emotion recognition robot for nursing environment. The robot monitors the daily activities and emotions of the residents, estimates the social relationships of the residents, takes care of the residents who are isolated from the community, and reduces their loneliness feelings by forming a good relationship in community.

Introduction

According to the United Nations, the population aging is progressing all over the world (United Nations 2015). As of September 2013, Japan became a super aging society where one quarter of the population is aged 65 years or older (Ministry of statistics 2013). Consequently, from April 2000 to April 2013, the number of nursing home residents has increased 1.71 times (Ministry of Health, Labor and Welfare 2014). In the survey of personnel shortfall for nursing home in 2017, 62.6% of nursing home answered that the caregiver was short (Person nursing labor stability center 2016). Despite the fact that the number of care recipients is increasing, care workers are in short supply. Therefore, the government encourage to introduce ICT in nursing environment to mitigate the workload of caretakers. In this context, many types of monitoring system are developed for nursing home. However, the monitoring system in the nursing home in the past is many in the bedroom and the area to be monitored is narrow. The ideal monitoring system monitors the entire nursing home such as the bathroom, the toilet, the dining room, the discourse room as well as the bedroom.

The dining room and discourse room can be accessed during the day and anyone can use it. Sadly, there exist some residents who spend most of their time in the bedroom without participating in the social activities such as television appreciation, conversation and recreation in the lounge, etc. These people are suffered from social isolation, which leads to many harmful health conditions (Nicholas 2012), which is becoming a problem not only in Japan but also worldwide. The social isolation is caused by various incidents, such as physical weakness, physical disorder, mental illness, deterioration of psychological function, social loss, and when these occur simultaneously, the elderly people often face to a deep social isolation (Karatsu 2012). It is also involved in depression, which is caused by a big life event such as relatives disappearing and chronic stress (Ministry of Health 2009).

We consider that creating a good relationship in the community is a useful mean to prevent social isolation. By creating a good community relationship, social activities are increased such as dining with others, chatting, playing game, watching TV together, helping others etc., which, in turn, decreases the loneliness feeling.

In this paper, we define Social Activities of Daily Living (SADL). Compared with ADL and Instrumental ADL (IADL), the former focuses on person's physical self-care abilities to perform independent living. And the latter focuses on person's mental-involved complex activities, SADL focuses on the social activities to measure the person's sense of community. Then, we propose a multimodal sensor-based recognition method for SADL, which is implemented in the IoT-based emotion recognition robot for nursing environment. The robot first recognizes person's activities and change of emotions by integrating the sensors' data: microwave sensor for vital data, video camera for the facial expressions, microphone for speech tone, environmental sensors for temperature, and brightness etc. Then, it evaluates the ADL, IADL index. SADL is also evaluated by monitoring the social relationships of the residents. The robot takes care of the residents who are isolated from the community by chatting on health-related talk. It also advises to create a good relationship in the community.

Related Work

Relationship between Loneliness and ADL

Relationship between loneliness or social isolation and various patterns of daily living are investigated(Goonawardene et al. 2017). In their research, Activity of homebound elderly people are analyzed. It is shown that the time spent in the living room is significantly correlated with the emotional loneliness. This suggests that single living elderly person who spends most time in living room feels lonely. Correlation of daytime napping duration with social loneliness is also analyzed, suggesting that if elderly people lack of sense of community, they sleep more during day time. From these studies, it can be considered that social isolation and loneliness can be measured by actively in daily living (ADL). Furthermore, this study shows that depression in the elderly people correlates with the loneliness and social isolation.

Emotion Recognition

Emotions can be recognized from various things such as facial expressions, voice, sentences, body temperature and so on. In patented technology by Panasonic (Panasonic), we recognize feelings form talking content and sounds of voice. In addition, the Empath API (Smartmedical) recognized feelings using the physical sound of speech. Microsoft's Emotion API (Microsoft)can recognize emotions from facial expressions. Recognition of facial expressions is also implemented as a function in the Omron camera module(Omron) that is used in our system.
Techniques for recognizing emotions like these are actively researched and these are applied in various fields.

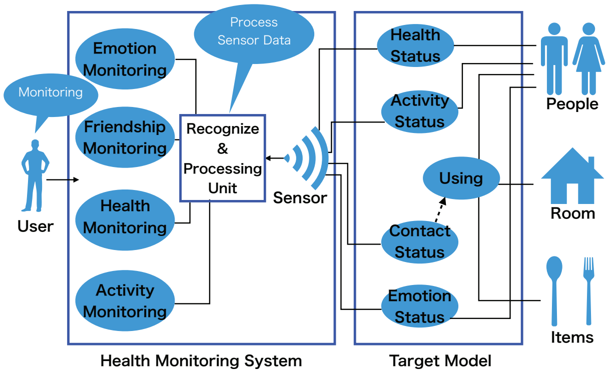


Figure 2 use case diagram

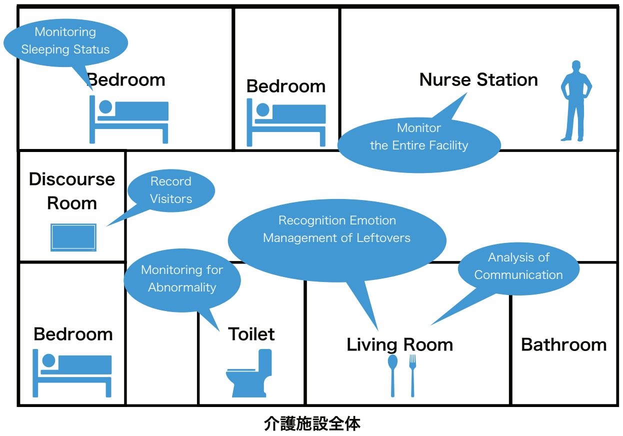

Monitoring Sleeping Status
Bedroom
Bedroom
Nurse Station
Monitor the Entire Facility
Discourse Room
Record Visitors
Recognition Emotion Management of Leftovers
Analysis of Communication
Monitoring for Abnormality
Bedroom
Toilet
Living Room
Bathroom
介護施設全体

Figure 1 Ideal Monitoring System

Proposed Monitoring System

The monitoring system in nursing home needs to be widespread including dining room and discourse room. The use case diagram of the monitoring system is shown in the Figure 1. "People", "Room" and "Items" on the right side indicate objects to be monitored. In the Target Model, it refers to the state to be monitored. The "Sensor" in the Health Monitoring System reads the information in the Target Model and recognizes what kind of state it is. "User" is the person who actually uses the monitor system. If you look at these in a nursing home, "People" are care-requiring persons who use a nursing home, "Room" is a room of a nursing home such as a bedroom, a living space, and "Items" is toothbrushes, dishes used for meals, furniture, and so on. "User" corresponds to a caregiver or a doctor. In Based on this use case diagram, we propose a monitoring system targeting the entire facility including the living space used by multiple people. Illustration of a desirable monitoring system in our proposal is shown in the Figure 2. It monitors the entire facility, including bedrooms that are often found in the conventional products. In addition to abnormality detection, we propose a system that utilizes information such as behavior recognition, conversation and expression, which leads to enhance community and living in the nursing home of the elderly. In this paper, we describe the implementation, experiment and evaluation of the IoT based robot with multiple sensors, which is developed to realize the proposed functions.

Proposed Model of Loneliness

In this research, we aim to measure the loneliness of the elderly automatically by the system, which in turn acts to support a creation of a good sense of community. The loneliness depends on several things, such as mental process and physical process. So, we propose to measure the loneliness of elderly people in three axes: physical loneliness, mental loneliness, and social loneliness. Physical loneliness is simply based on whether or not there are people around the elderly. Mental loneliness is measured from emotions such as expression. Also, actions such as reading and talking are related to mental loneliness. Social loneliness is measured by the amount of social activity such as recreation.

Model of Loneliness

Calculation of the loneliness of each of the three axes is performed by recognizing the actions in daily living. Consideration is given to the relationships between typical actions in daily living and the physical loneliness, mental loneliness and social solitude. Table 1 gives examples of points of senior citizens' behavior and loneliness levels related to them.

For example, consider the act of eating with others in the dining room at meal time, that action is considered to be beneficial to reduce the physical loneliness and social isolation, thus the good points are given to these two axes. The higher the score of each axis, the lower the loneliness of elderly people feel.

Define a table that combines such actions and points for typcical activities in daily living.

Table 1 relationship between behavior and loneliness

Behaviors	Physical	Mental	Social
Eating with everyone	+1	+1	+1
Speak using cell phone	+0	+1	+0
Visit discourse room	+0	+0	+1
Join Recreation	+1	+1	+1

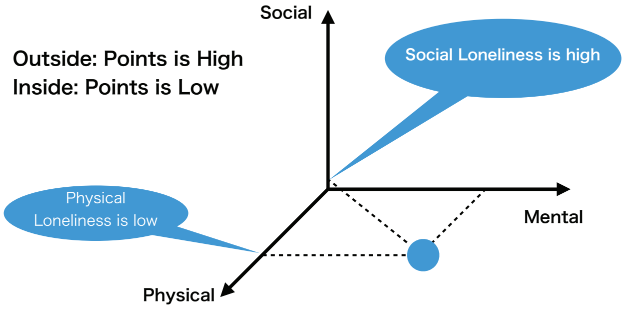


Figure 3 Mapping of Loneliness

Mapping of Loneliness

Using the table shown in Table 1, we can summarize the all activities in daily living and map the results into the three axes shown in the Figure 3 as the degree of loneliness in a day.

By analyzing this three-dimensional figure, it is possible to grasp the tendency of which axis the solitude degree is high or low. After mapping, we can make a personalized care plan to reduce the solitude which is different peron by person.

Monitoring System Design and Implementation

System Design

Functional requirements of the monitoring system are as follows:

(1) Status Monitor

The most basic function is to monitor person' various status such as position, posture, activity, and vital sign. Different kinds of sensors should be used, thus integration of sensor data to reason higher level status is necessary.

(2) Emergency Watch

If the person is in a critical status, such as stroke and falling, the system should recognize the status and put alert within a specified time, such as 1 minitute, that means the system's response time requirement is essential.

(3) Forecasting

Forecasting of status change is also desirable. For daily living, prediction of wake up time or urination is important for a caregiver.To realize it, the system should store the activity log and analyze to find a daily life pattern.

(4) Dialog

Normal functions of monitoring are passive. The monitoring should be performed without bothering the residents. But for isolated elders, active monitoring function should be considered, such as dialog and greeting. The reactive planning capability is necessary to realize it.

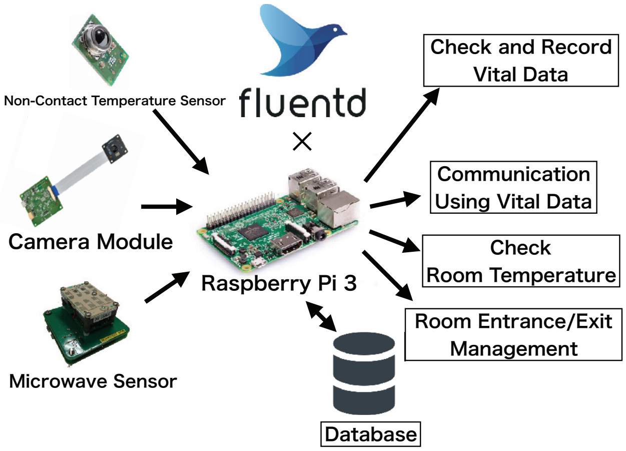

Non-Contact Temperature Sensor
fluentd
×
Camera Module
Raspberry Pi 3
Microwave Sensor
Database
Check and Record Vital Data
Communication Using Vital Data
Check Room Temperature
Room Entrance/Exit Management

Figure 4 System Configuration

Configuration

The system configuration diagram is shown in the Figure 4. All sensors are mounted on RaspberryPi3: camera module, non-contact temperature sensor, microwave sensor. We use Fluentd S/W toolkit which is originally developed for a Web server log collection, for our data-driven architecture. It can processes and integrates data flowing asynchronously from each module, and utilizes them for recording of vital data, communication by utterance function, and behavior recognition. The log data and the name of the elderly are recorded in the database.

Experiment and Evaluation

Experiment Method

We actually installed a IoT based robot in a nursing home, and conducted an experiment. The purpose of the experiment is two-fold.

First, since the camera module, the non-contact temperature sensor, and the microwave sensor operate asynchronously, it is necessary to integrate them and perform simple communication with the elderly using the integrated data. The verification of this basic function should be checked. The communication performed this time is to detect the face of a person and speak with data of heart rate, respiratory rate, expression, obtained from each module if it is a person registered in the database. Figure shows implemented scenarios. This scenario assumes that the robot will greet the elderly. If there is only one face detected, speak the name, the expression at that time, and vital data. If two or more people, call their name and greet. In this scenario, the robot uses camera module and microwave sensor for detecting face and sensing the heart rate and breathing rate.

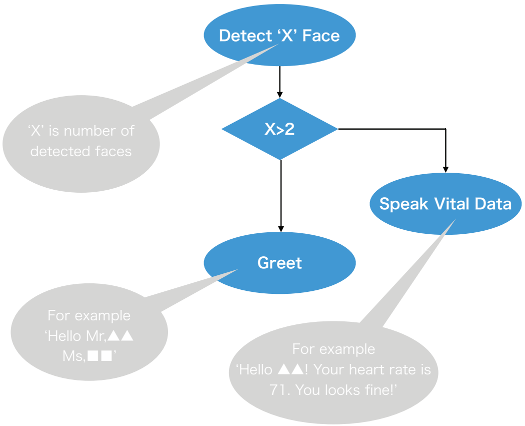

Detect 'X' Face
'X' is number of detected faces
X>2
Speak Vital Data
Greet
For example 'Hello Mr.▲▲ Ms.■■'
For example 'Hello ▲▲! Your heart rate is 71. You looks fine!'

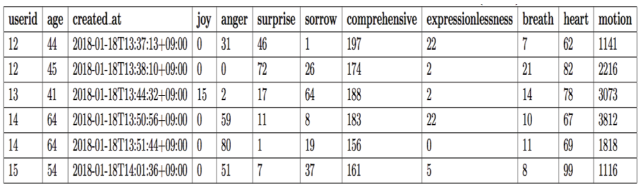
Figure 5 Implemented Scenario

The second is to recognize one SADL, "visit a discourse room" and to measure the social loneliness degree of elderly people in a simple way. For recognition of this SADL, a camera module is used. When the robot monitors the entire discourse room and recognizes the face, it records the ID of that person and the time visited in the database. Count the number of visits by elderly people and measure the social loneliness with the number of visits as a score. For the experiment, we registered the face of four facility users in the robot. The experiment was conducted for about 2 days from 13:00 on January 18, 2018 to January 19, 2018.

Results of Face Recognition

After registering the face of the person and verifying it in the discourse room, The IoT-based robot could communicate with the elderly using sensor data. Table 2 shows a part of vital data such as the heart rate of the user actually obtained. Even the elderly talked to the robot, and many laughing facial expressions were observed. Figure 4 shows the state of the experiment at the actual nursing home.

Table 2 Vital Log Data



userid	age	created_at	joy	anger	surprise	sorrow	comprehensive	expressionlessness	breath	heart	motion
12	44	2018-01-18T13:37:13+09:00	0	31	46	1	197	22	7	62	1141
12	45	2018-01-18T13:38:10+09:00	0	0	72	26	174	2	21	82	2216
13	41	2018-01-18T13:44:32+09:00	15	2	17	64	188	2	14	78	3073
14	64	2018-01-18T13:50:56+09:00	0	59	11	8	183	22	10	67	3812
14	64	2018-01-18T13:51:44+09:00	0	80	1	19	156	0	11	69	1818
15	54	2018-01-18T14:01:36+09:00	0	51	7	37	161	5	8	99	1116

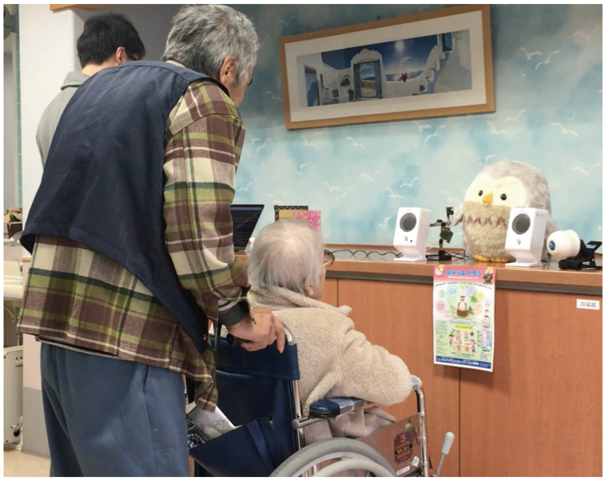


Figure 6 The robot is talking to user

Results of SADL Recognition

It was possible to obtain the time and the staying time of visiting the lounge room of the subject whose name was registered in the database. The Figure 5 shows the result of recognition of SADL. The horizontal axis is time, which is a point indicating that a round plot has visited the discourse room. The top one is the log data of user 1, her visiting was at 14:22 on 18th and never on 19th. Likewise, user 3 never visited the lounge on the 19th. On the other hand, user 2 and user 4 were visiting the lounge room on the 19th.

Based on Figure 5, Table 2 shows how many times each user visited the lounge room manually. If the intervals of the plots are short, it is assumed that they are recognized multiple times by one visit, and a set of plots in which one hour or more is free is taken as "one visit". Based on the proposed Model of Loneliness, we will measure the loneliness of each user. Assuming that the behavior of visiting the discourse room is related only to the degree of social isolation, it can be said that user 1 has the most social degree of solitude among the four.

Table 3 User Visits

User	User Visits
User1	2
User2	5
User3	4
User4	4

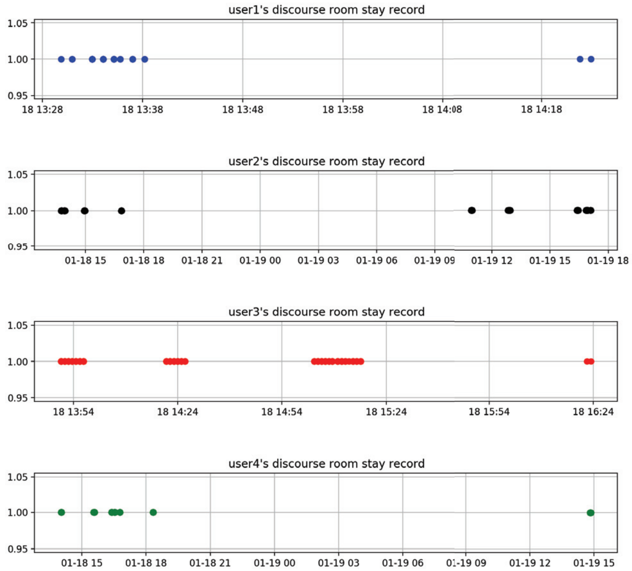


Figure 7 Visitor to discourse room

Discussion

In this experiment, we are only able to observe the visitor in the discourse room, we cannot map the social degree of loneliness because we cannot currently recognize other actions. By increasing the recognizable behavior, it is possible to measure significant degree of loneliness. In order to increase the number of recognizable behaviors, it is necessary to handle not only the sensor built into the robot but also the data of the remote sensor installed at a location away from the robot, such as a bedroom or dining room. By doing so, the scope of monitoring as a monitoring system will also become wider, and it will be possible to increase the number of types of behaviors that can be recognized and to be able to map loneliness to significant 3 axis diagrams at the same time.

Moreover, by using the vital data obtained by the experiment and the recognition result of SADL for visualization and measuring the degree of loneliness, it is expected to realize a system which can support to reduce the caregivers' work load.

It is thought that an interface using speech recognition technology becomes essential. When we were experimenting at a facility, we often saw robots are talked back. we actually got an opinion from the residents that "I wat to continue conversation after the robot talked to me".

Since speech recognition can be used by a caregiver who both hands are occupied in doing work, we will also consider application in the field of ICT to convey work efficiency and information on the elderly.

Conclusion

In this paper, we propose a next-generation monitoring system in a nursing home, and developed a monitoring robot which implemented a part of the desirable functions. We also proposed a model for measuring elderly loneliness. The robot incorporates multiple sensors and a speaker so that communication can be taken.

As a result of the experiment using the robot, it was possible to consolidate the sensor data sent to fluentd asynchronously and to store the log data such as the expression of the elderly at that time in association with the person. We were able to communicate easily with the elderly people. Also, we were able to recognize SADL, by monitoring who and when the visitor comes. By using the result of recognizing SADL, we could measure social loneliness degree of elderly person in a simple manner by counting the number of visiting times in the discourse room. Currently, recognizable behaviors are limted. Although few measurements of the degree of solitude have been made, future data of the remote sensor will be utilized to increase the recognizable behavior.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP17H01823 “Development of Watching System by Integration of Non-Restrictive Sensors”. We would like to thank WCL and Long-Term Care Health Facilities ‘Yui’ for their support. Furthermore, we would like to thank Masayuki Numao giving valuable comments on the manuscript.

References

United Nations Department of Economic and Social Affairs. 2015. World Population Ageing.

Ministry of statistics. 2013. Population of elderly people.

Ministry of Health, Labor and Welfare. 2014. The situation surrounding the long-term care insurance system.

Person nursing labor stability center. 2016. Result of nursing care labor survey.

Nicholas R. Nicholson. 2012. A Review of Social Isolation. The Journal of Primary Prevention, 33(2-3):137-152.

Hiroshi Karatsu. 2012. A Study on Elderly People’s Isolation in the Super Aging Society (in Japanese). Nara Bunka Women's Col-lege. pp189 (in Japanese).

Ministry of Health, Labor and Welfare Report. 2009. *Basic knowledge of depression of the elderly (in Japanese)*. http://www.mhlw.go.jp/topics/2009/05/dl/tp0501-siryou8-1.pdf

Panasonic Corporation. *PASTA*. Retrieved November 8, 2017, https://feeling.pas-ta.io/.

Smartmedical Co., Ltd.. *Empath API*. Retrieved November 8, 2017. https://webempath.net/lp-jpn/.

Microsoft Corporation. *Emotion API*. Retrieved November 8, 2017. https://azure.microsoft.com/ja-jp/services/cognitive-services/emotion/.

OMRON Corporation. *HVC-P2*. Retrieved November 8, 2017. http://plus-sensing.omron.co.jp/product/hvc-p2.html.

Active Online Learning Architecture for Multimodal Sensor-Based ADL Recognition

Nobuyuki Oishi, Masayuki Numao
Department of Communication Engineering and Informatics
The University of Electro-Communications
n.oishi@uec.ac.jp, numao@cs.uec.ac.jp

Abstract

Long-term observation of changes in Activities of Daily Living (ADL) is important for assisting older people to stay active longer by preventing aging-associated diseases such as disuse syndrome. Previous studies have proposed a number of ways to detect the state of a person using a single type of sensor data. However, for recognizing more complicated state, properly integrating multiple sensor data is essential, but the technology remains a challenge. In addition, previous methods lack abilities to deal with misclassified data unknown at the training phase. In this paper, we propose an architecture for multimodal sensor-based ADL recognition which spontaneously acquires knowledge from data of unknown label type. Evaluation experiments are conducted to test the architectures abilities to recognize ADL and construct data-driven reactive planning by integrating three types of dataflows, acquire new concepts, and expand existing concepts semi-autonomously and in real time. By adding extension plugins to Fluentd, we expended its functions and developed an extended model, Fluentd++. The results of the evaluation experiments indicate that the architecture is able to achieve the above required functions satisfactorily.

Introduction

The world's elderly population is growing rapidly, particularly in more developed countries where longevity and the increase of elderly people seem to impact negatively on the proportion of the working age population. This situation causes serious social issues and problems including medical staff shortage for homecare services. To deal with the problems, Ambient Assisted Living (AAL) is gaining a great deal of interest. AAL uses IoT and AI technology to support to support older people to live independently for as long as possible and improve their quality of life. In the AAL community, Activities of Daily Living (ADL) observation by IoT is one of the most attention-getting topics (Monekosso, Florez-Revuelta, and Remagnino 2015). For example, long-term observation of changes in ADL is one thing considered to be important for preventing disuse syndrome. Once older people suffer from the disuse syndrome, it is difficult to ameliorate their conditions to the original level. Therefore, giving older people some awareness of how active they are becoming by checking older peoples ADL and preventing the disuse syndrome could be more important than treating it.

Many of the previous studies have proposed various approaches to detect the state of a person by using a single type of sensor data. However, single type of sensor cannot provide enough information, thus it is essential to use multiple sensors for recognizing more complicated state, but integrating different kinds of data from multiple sensors are difficult and the integration technology is not well established. Also, previous methods lack the ability to deal with data of unknown label type. It means that they do not possess enough capability to deal with unexpected actions or conditions. This paper proposes an Active Online Learning (AOL) architecture that can semi-autonomously acquire knowledge from unknown label type data for multimodal sensor-based ADL recognition.

Sensor-based Activity Recognition

Currently, various kinds of sensors, i.e. infrared motion sensors, accelerometers and cameras, are available for ADL measurements. In addition, RFID is one of the most widely used technologies for human activity recognition, and it usually uses small passive tags which don't require an internal power source, thus they can be easily attached to person's wears and tools. The position and attitude of the tagged object can be calculated by RSSI. Among numerous studies on sensor-based activity recognition, Fortin-Simart et al. have demonstrated that several ADLs in the kitchen such as making coffee and getting a bowl of cereal can be distinguished with higher accuracy than 90% (Fortin-Simard et al. 2015).

Related Work

Multimodal Learning

Information in the real world is provided as multimodal information such as text, image, and audio. Humans recognize things by ingeniously integrating those modalities. In the same ways, multimodal machine learning uses multimodal information to recognize a more complicated situation and do more complicated tasks than when single modality is used. Based on a taxonomy proposed by Tadas B. et al., there are mainly five challenges that multimodal learning faces: representation, translation, alignment, fusion and co-learning (Baltrusaitis, Ahuja, and Morency 2017). Out of these five challenges, fusion is the one that this paper addresses in order to join information from two or more modalities to perform a prediction. Libal V. et al. have conducted an experiment to classify 6 ADLs, i.e. eating-drinking, reading, ironing, cleaning, phone answering, and TV watching, by using microphones and cameras installed inside an apartment. Their results show that classifier using simple concatenative features of audio and visual significantly outperformed both unimodal classifiers: 65.97% when using audio-visual features, 57.64% when using only an audio feature, 46.53% when using only a visual feature (Libal et al. 2009).

Active Learning

Active learning is a subfield of machine learning. The key idea behind active learning is that greater accuracy can be obtained with fewer labeled training instances if a machine learning algorithm is allowed to choose the data to learn (Settles 2009). An active learner can request an oracle to label unlabeled instances. Active learning is useful where there is already a large amount of data or unlabeled data can be easily obtained and there is high labeling cost. As recognizing ADLs by using sensor data deals with large amount of data and unlabeled data as well as high labeling cost, active learning is considered to be an effective approach for recognizing ADLs. Sensor data itself can be easily obtained once sensors have been installed. However, labeling them for a computer to learn is very laborious. If an active learning algorithm can find which instances are most informative and then ask an oracle to label them, active learning approach can solve the labeling problem.

Online Learning

Online machine learning can sequentially update its predictor for future data as training data comes while batch learning fixes its predictor once learning phase with training dataset has been done. This property makes it possible for a machine learning algorithm to maintain and improve its model so as to adapt to changes in an environment. Masuda and Numao have proposed a real-time human state detection system and has confirmed that the detection model can keep high accuracy rate more than 90% by additionally trained detection model with data collected from different subjects in a setting where a detection model trained with data collected from a single subject is used for another subject (Numao and Masuda 2016).

AOL Architecture for ADL Recognition

In this paper, we propose an architecture for multimodal sensor-based ADL recognition. This architecture is able to recognize ADLs based on data-driven approaches and semi-autonomously acquire knowledge about activities of daily living. Figure 1 provides an overview of the Active Online Learning (AOL) architecture. This architecture's functions can be largely separated into two parts:

1. ADL recognition
2. New concept learnering

ADL Recognition in Three Types of Dataflows

ADLs are recognized based on data flown in the following three types of dataflows: sensor-level dataflow, action-level dataflow, and concept-level dataflow. Each dataflow has its own role and cooperates each other to recognize ADLs and utilize utilize obtained results.

Sensor-level Dataflow A large quantity of data obtained from various kinds of sensors is sent to the sensor-level dataflow. The role of this dataflow is to process the data into easy-to-use form by extracting important information or merging multiple data according to pre-determined rules so that subsequent processes can be easily done. It includes detecting some ADLs based on simple if-then rules e.g. detecting entering/leaving a room when RSSI of an RFID tag attached to a person is higher/lower than a certain level. Then the processed data or recognized action data is sent to next dataflows.

Action-level Dataflow The action-level dataflow receives recognized data of ADLs from the sensor-level and concept-level dataflows. Then its reactive planning engine processes the recognized results and gives a command to the activator of the intelligent caretaker to do some tasks such as speaking out generated texts according to the obtained data or executing speech recognition program to get a response of a person whom the intelligent caretaker is interacting with. This role of the reactive planning engine is very important for this architecture because it determines what to do with recognized results obtained from the sensor- and concept-level dataflows. This is an example of assumed scenarios: If one can track peoples location and posture using RFID or some other sensors, the system can give a warning when someone enters the toilet and does not move for over 15 minutes.

Concept-level Dataflow The concept-level dataflow recognizes ADLs using online machine learning techniques and sends the estimated results to the action-level dataflow. While the sensor-level dataflow is in charge of relatively low-level ADL recognition such as just applying simple if-then rules to the obtained data, the concept-level dataflow is responsible for more advanced ADL recognition tasks.

Also, it keeps updating its activity recognition model as associating with the new concept learning function. When it encounters new patterns in the data sequence, it notifies that it has detected new patterns and updates its activity recognition model with the data and its correct label given by an oracle. It is also expected to detect unexpected anomalous actions or conditions.

Data-Driven Reactive Planning

Reactive planning is a different approach concerning with the problem of planning under uncertainty (Pryor L. 1996). It denotes a group of techniques for action selection by autonomous agents that operate in a timely fashion and compute just one next action in every instant, based on the current context (Reviews 2016).

This study utilizes similar techniques that enable the algorithms make decisions based on the results of data recognition within a very short time (e.g. 0.1 sec.), real time and

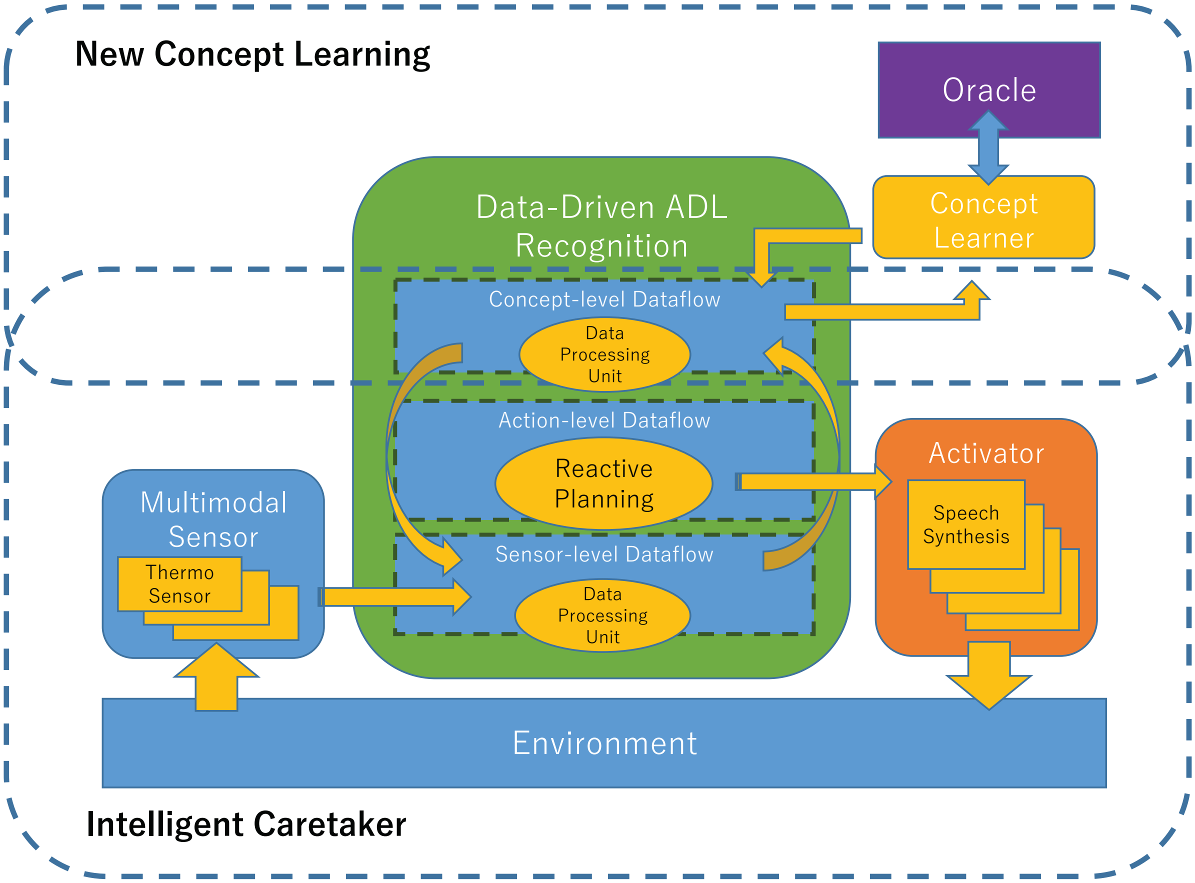


Figure 1: Overview of the active online learning architecture for ADL recognition

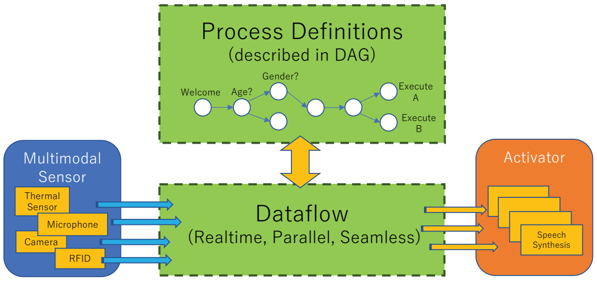


Figure 2: Overview of Data-Driven Reactive Planning

seamlessly. The baseline of the proposed architecture sets the data to be parallel and actions to be sequential in a process described in Directed Acyclic Graphs (DAGs) (See Figure 2)

New Concept Learning

New concept learning is one of the most important key feature of this proposed architecture. This function adds abilities of learning new concepts and expanding existing concepts to this architecture. The data-driven ADL recognition passes uncategorized data to the new concept leaner when it encounters data which is unlikely to be classified into an existing class, and then the new concept learner requests correct label corresponding to the received data to an oracle. Next, the new concept learner updates its activity recognition model with the data and its correct label given by the oracle. When the given label is a new label for the activity recognition model, it means that new concept will be added to the candidate class list, and when the given label already existing, it means that the existing concept will be expanded. By repeating this process, asking a correct label to the oracle and updating its activity model, the online learning classifier in concept-level dataflow becomes smarter and smarter and to be able to recognize more complicated activities with higher accuracy. Figure 3 shows the process of acquiring a new concept.

Intelligent Caretaker

The intelligent caretaker is a smart robot which has the data-driven AOL architecture inside of it. It collects data from various sensors mounted in the robot's body or placed in the environment and feedback to the external environment according to obtained results. The intelligent caretaker is in charge of interacting with the external environment. It is also expected to helps the system give a sense of intimacy to older people.

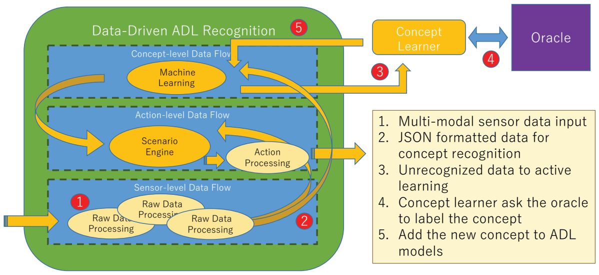

Data-Driven ADL Recognition
5
Concept-level Data Flow
Machine Learning
Action-level Data Flow
Scenario Engine
Action Processing
Sensor-level Data Flow
1
Raw Data Processing
Raw Data Processing
Raw Data Processing
2
Concept Learner
3
Oracle
4
1. Multi-modal sensor data input
2. JSON formatted data for concept recognition
3. Unrecognized data to active learning
4. Concept learner ask the oracle to label the concept
5. Add the new concept to ADL models

Figure 3: New concept acquisition flow

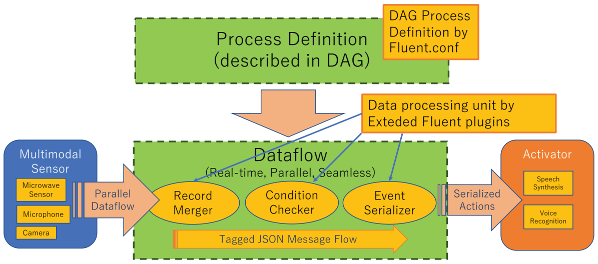


Figure 4: Fluentd++

Implementation

We have implemented the proposed architecture using open source log collection software Fluentd for data-driven data processing in the architecture, Raspberry pi 3 for the intelligent caretaker, and open source online distributed machine learning platform Jubatus respectively.

Three Dataflows in Fluentd

Data flown in Fluentd is structured as JSON format so that we can easily unify log data across multiple sources and destinations. Also, Fluentd has a very flexible plugin system that allows users to extend its functionality depending on their specific needs. Hence, Fluentd can be considered that it is not just a log collection software but also a very useful data stream management tool which let us apply data-driven approach.

A Fluentd event consists of a tag, time and record. A tag has period-separated name structure (e.g. sensor.temp_hum.in_living_room) and is used for routing Fluentd events. The time shoes when the event came and the record has its JSON formatted data. Following is an example of a Fluentd event:

```
2017−11−03 19:35:32 +0900 sensor.microwave:
{”heart”:79, ”breath”:12}
```

We have determined tag naming conventions. By using its tag name-based routing rule, the three kinds of dataflows, sensor-, action-, and concept-level, can be distinguished internally such as ”sensor.” for sensor-level dataflow, ”action.” for action-level dataflow, and ”concept.” for concept-level dataflow.

Fluentd++ for Data-Driven Reactive Planning

As we encountered difficulties in handling asynchronous data from the sensors using Fluentd during testing phase and in defining processing scenarios, it was necessary to expand the functions of Fluentd. To do so, we made additional plugins: record merger plugin to integrate multiple events arriving at different timing, condition checker plugin to make it easier and simpler to apply if-then rules to the data stream, and event serializer to sequence the multiple events detected. By installing the three plugins, which default Fluentd does not possess, we were able to handle the parallel data more accurately and efficiently as well as flexibly and achieve our goals of activity recognition, such as face and emotion recognition, and reactive planning, such as requesting for smile or measuring detected person's vitaldata. Figure 4 is an overview of the Fluentd++.

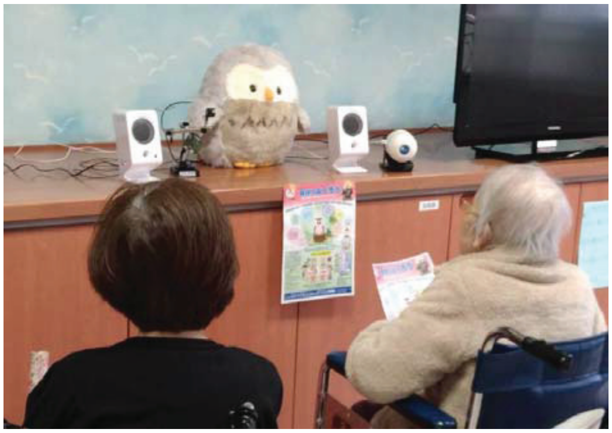


Figure 5: Caring Owl talking to women

Intelligent Caretaker

We have assembled an intelligent caretaker named “Caring Owl” together with WCL Co., Ltd. using Raspberry Pi 3. Fluentd++ is installed and the data-driven ADL recognition architecture is running on it. It is equipped with several sensors including a face recognition camera, a microwave sensor, a non-contact thermal sensor, a temperature-humidity sensor, and a microphone. It can also collect data from sensors placed in the external environment such as a mattress sleep sensor through the wireless network. Obtained sensor data can be easily sent to Fluentd by using an event sending interface fluent-logger. It is available for most major programming languages. We have started conducting an experiment with the Caring Owl in a nursing home. Figure 5 shows the scene that the Caring Owl is talking to women.

New Concept Learnering

To do this, active learning plugin for fluentd++ were made. It adds function to issue queries to an oracle when unclassifiable data arrives and to get new concepts by using received data from the oracle. In more detail, when the online learning classifier in concept-level dataflow encounters data which is unlikely to be classified into an existing class, it outputs the data to a specified filepath. The new concept learner keeps watching the output file and executes the following each time when it is updated.

1. When new unclassifiable data comes, check the elapsed time since the last unclassifiable data came; and if the time lag is exceeded specified time limit, delete all the data in the log file except the latest one; but if the time lag is shorter than the time limit, just append the new unclassifiable data at the end of the log file
2. If the specified amount of unclassifiable data has accumulated in the log file, request the correct label corresponding to the data to an oracle
3. Subsequently, update the activity model using the accumulated data and the label

There are two reasons why a certain amount of unclassifiable data is stored in a log file. The first reason is that it is just bothersome and inefficient to do labeling every time. The second reason is that sensor data often contains noises and labeling noisy data is waste of energy or even harmful to the activity recognition model.

Evaluation Experiments

In this experiment, we had mainly two kinds of experiments: one was about whether the architecture implemented with Fleuntd++ can (1) actually describe reactive planning and work as expected; the other was whether this architecture can (2) acquire new concepts and (3) expand existing concepts adjusting its activity model according to individual differences, and (4) how much the labeling costs for each case are.

Experiment 1: Data-Driven Reactive Planning with Fluentd++

By using Fluentd++, we defined scenarios to be Figure6, confirmed the actions and conducted performance evaluation.

- Scenario 1: Face recognition is done by the face recognition camera. The Intelligent Caretaker sends greetings when the faces of two or more pre-registered people are detected at the same time.
- Scenario 2: Face recognition camera detects pre-registered faces individually. When a recognized face is in the center, the Intelligent Caretaker greets the person and stores the persons vital data (body temperature, heart rate and respiration rate) taken from the sensors into the database.
- Scenario 3: Face recognition camera detects pre-registered faces individually. When the recognized face is not in the center, the Intelligent Caretaker requests the person to move to the center in order to properly obtain the persons vital data from the sensors.
- Scenario 4: In the case that the face recognition camera recognizes the face(s) but finds no pre-registered faces and that there are more than 5 smiling faces, the Intelligent Caretaker greets the people, “You look like you are having fun!”

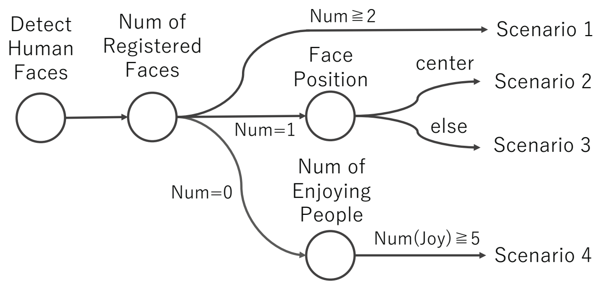
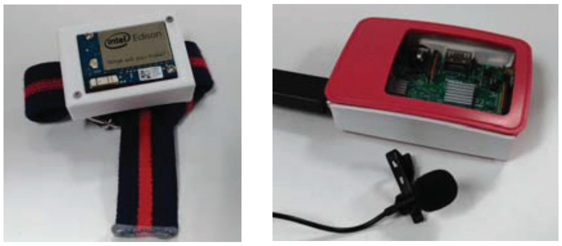


Figure 6: DAG based Process Definition



[1] 9DoF sensor + Edison [2] Pin mic + RaspPi

Figure 7: The 9DoF sensor and the pin mic

Fluentd++ outputs all the data flown on dataflow to log files. We evaluated how much data were processed and the real-time response to the recognized results.

Experiment 2 and 3: Data Collection for AOL

We used a 9DoF(Degrees of Freedom) sensor (3-axis accelerometer, 3-axis gyro, and 3-axis magnetometer) and a pin mic, wearing on the dominant wrist and on the chest respectively. The 9DoF sensor is attached to an Intel Edison and the pin mic is connected to a Raspberry Pi 3 so that they can send collected data through a wireless network. Collected data are converted into feature vectors and then sent to the data-driven ADL recognition unit roughly every second which is running on an intelligent caretaker.

For the 9DoF sensor, a value of each axis is obtained every 0.1 seconds and mean and variance values of them are calculated every second and used as feature vectors. For the pin mic, MFCC(Mel Frequency Cepstral Coefficients), RMS(Root Mean Square), and ZCR are calculated every second and used as feature vectors. All the feature vectors are normalized between -1 and 1 using min-max normalization method.

Experiment 2 and 3: AOL Algorithm

The k-Nearest Neighbor(k-NN) algorithm is used as a classification algorithm in the concept-level dataflow in the ADL recognition unit, and Locality Sensitive Hashing(LSH) based on cosine similarity is used for finding nearest neighbors. We set k = 10. Obtained data would be treated as uncategorized data when the max score is smaller than 9.5. The score for each class is calculated from following equation:

$Score(class) = exp(-distance * S)$ S is the sensitivity with respect to distance, which is set to 0.1. The procedure of issuing queries to an oracle follows described in Implementation of New Concept Learning. The time gap limit is set to 3 seconds and when 10 unclassifiable data has accumulated, the new concept leaner requests the corresponding label to the data.

Experiment 2: Acquiring New Concepts

Firstly, we tried to find out whether this architecture can actually acquire new concepts spontaneously. We collected sensor data from a subject for 10 seconds each of the following five classes and train the NN-classifier in advance.

- Sitting
- Standing
- Walking
- Vacuuming
- Hair-drying

We call the subject subject-A in order to distinguish it from the other subject, and we call the other subject subject-B. After finishing the pre-training, we kept repeating the five activities above and answered questions issued by the new concept leaner until it stopped asking a question. Then, we began to do the following five activities.

- Washing hands
- Brushing teeth
- Gargling
- Speaking
- Stairs up/down

If the new concept learner works properly, it would request new labels corresponding to these activities and learn these activities as new concepts. We counted how many times the new concept leaner needed to issue requests for each label. Also, we classified subject-A's activities using the trained model and analyzed the result.

Experiment 3: Expanding Existing Concepts

Secondly, we tried to find out whether this architecture can actually expand existing concepts according to individual difference. The activity recognition model was only trained by subject-A's behavior, which meant that this model might not work well on subject-B's activity classification. However, as well as acquiring new concepts, the new concept learner would request labels when ambiguous data arrived and try to adjust its model with true labels given by an oracle.

We compared the accuracy rate between the following cases, (a) one is prohibited to request true-label to an oracle and (b) the other is allowed. Also, we counted how many times the new concept leaner needed to ask a question for each label in the latter case.

Results and Analysis

Data-Driven Reactive Planning with Fluentd++

First, the total numbers of sensor data on dataflow and bytes within the 30 hours of system operation are 277596 and 59.4 MB. On average, 2.6 numbers of data were sent to dataflow. The maximum number of data per second counted was 197 (inclusive of the error messages emitted when the connection with MQTT broker was broken and of the normal sensor data). The numbers of execution of each scenario are obtained and results are as follows: $\{$Scenario 1: 1, Scenario 2: 97, Scenario 3: 40, Scenario 4: 1$\}$. Next, we evaluated the performance of the system construction. Within the 30 hours of experiments, the number of execution of speech scenarios by the Intelligent Caretaker was 139. For all the 139 times of scenario execution, Fluentd++ was able to process all the scenarios triggered by the face recognition camera within 1 second each time. (With the current settings, we were unable to measure the time in millisecond scale or to have more precise time than seconds.) In addition, we confirmed that action serialization functioned normally, i.e. being able to sequence the multiple events chronologically.

Acquiring New Concepts

Table shows that how many times the new concept learner requested and updated its ADL recognition model. The largest number of input times was 6 for going up/down the stairs. This shows that at most several times of label input operation is sufficient for learning one class of activity. A reason why seemingly easy activities such as sitting or standing required relatively large numbers of labeling request are considered that 3-axis geomagnetic sensor's values are used as a part of the feature vector and it put an importance on which direction the subject was looking. For the same reason, activities such as washing hands and brushing teeth which are taken in a limited position required just a few times of learning.

The confusion matrix in Table 2 shows the performance of the obtained ADL classification model. Accuracy rate was over 90% in all classes except standing. Especially, it appears that the new concept learner properly has requested labels to the five latter activities and acquired new concepts corresponding to them.

Expanding Existing Concepts

Table 4 shows the performance of ADL classification of the following cases (a) directly using subject-A's ADL recognition model, (b) using updated subject-A's ADL recognition model by the new concept learner respectively. The numbers in brackets means the results of (b). Although accuracy rates of sitting and standing classes have mildly declined, those of gargling, speaking, and up/down the stairs classes got improved significantly. Table shows the numbers of label input times to train the subject-A's ADL recognition model in order to fit the subject-B's behavioral patterns. It was also assumed that just a few times of label input for each class is enough to adjust the model in this case. These results above show that this architecture can actually acquire new concepts

Table 1: Numbers of Label Input Times for Acquiring New Concepts

Actions Class	Sitting	Standing	Walking	Vacuuming	Blow-drying	Washing hands	Brushing teeth	Gargling	Speaking	Up/Down the stairs
Label Input Numbers	5	5	3	4	1	1	1	1	3	6

Table 2: Results of Experiment 2: Confusion Matrix

Activity	Classified As										Accuracy Rate
	Sitting	Standing	Walking	Vacuuming	Blow-drying	Washing Hands	Brushing Teeth	Gargling	Speaking	Stairs Up/Down	
Sitting	**37**	0	0	0	0	0	0	0	0	0	1.0
Standing	0	**21**	6	0	0	0	0	0	0	12	0.53
Walking	0	0	**55**	0	0	0	0	0	0	2	0.96
Vacuuming	0	0	0	**38**	0	0	0	0	0	0	1.0
Blow-drying	0	0	0	0	**40**	0	0	0	0	0	1.0
Washing Hands	0	0	0	0	0	**39**	0	0	0	0	1.0
Brushing Teeth	0	0	0	0	0	0	**39**	0	0	0	1.0
Gargling	0	1	0	0	0	0	0	**23**	0	0	0.96
Speaking	0	0	0	0	0	0	0	0	**40**	0	1.0
Stairs Up/Down	0	3	0	9	0	0	0	0	0	**116**	0.91

Table 3: Numbers of Label Input Times for Expanding Existing Concepts)

Actions Class	Sitting	Standing	Walking	Vacuuming	Blow-drying	Washing hands	Brushing teeth	Gargling	Speaking	Up/Down the stairs
Label Input Times	0	0	0	0	0	0	0	3	1	6

Table 4: Results of Experiment 3: Confusion Matrix

Activity	Classified As										Accuracy Rate
	Sitting	Standing	Walking	Vacuuming	Blow-drying	Washing Hands	Brushing Teeth	Gargling	Speaking	Stairs Up/Down	
Sitting	**42 (47)**	0(0)	0 (1)	0 (0)	0 (0)	0 (0)	0(0)	0 (0)	0 (0)	0 (0)	1.0 (0.98)
Standing	0 (0)	**26 (34)**	0 (7)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	1.0 (0.83)
Walking	0 (0)	0 (0)	**45 (45)**	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	1.0 (1.0)
Vacuuming	0 (0)	0 (0)	0 (0)	**47 (42)**	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	1.0 (1.0)
Blow-drying	0 (0)	0 (0)	0 (0)	0 (0)	**39 (38)**	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	1.0 (1.0)
Washing Hands	0 (0)	0 (0)	0 (0)	0 (0)	0(0)	**39(39)**	0 (0)	0 (0)	0 (0)	0 (0)	1.0 (1.0)
Brushing Teeth	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	**38 (35)**	0 (0)	0 (0)	0 (0)	1.0 (1.0)
Gargling	0 (0)	0 (1)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	**3 (13)**	37 (0)	0 (0)	0.08 (1.0)
Speaking	19 (10)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	**28 (24)**	0 (0)	0.60 (0.71)
Stairs Up/Down	0 (0)	1 (1)	35 (5)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	0 (0)	**17 (94)**	0.32 (0.94)

and expand existing concepts spontaneously and be able to classify ADLs with higher accuracy than 90% on average.

Conclusion and Future Works

We proposed an active online learning architecture for multimodal sensor-based ADL recognition for real-time recognition and learning to achieve better practicality. Also, we conducted three evaluation experiments to test the architectures abilities to recognize ADL and construct data-driven reactive planning by integrating three types of dataflows, acquire new concepts, and expand existing concepts semi-autonomously and in real time. By adding extension plugins to Fluentd, we expended its functions and developed an extended model, Fluentd++. The results of the evaluation experiments indicate that the architecture is able to achieve the above required functions satisfactorily. In this research, positive results were obtained from a simple combination of the 30 features calculated from multimodal sensor data. However, we are uncertain that the same results would be achieved when the number of features increases as a result of increased sensors. Therefore, our plan for future research is to improve the feature selection methodology.

Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP17H01823 “Development of Watching System by Integration of Non-Restrictive Sensors”. Also, we would like to thank WCL and Long-Term Care Health Facilities Yui for their support.

References

Baltrusaitis, T.; Ahuja, C.; and Morency, L. 2017. Multimodal machine learning: A survey and taxonomy. *CoRR* abs/1705.09406.

Fortin-Simard, D.; Bilodeau, J.-S.; Bouchard, K.; Gaboury, S.; and Bouzouane, A. 2015. Exploiting passive rfid technology for activity recognition in smart homes.

Libal, V.; Ramabhadran, B.; Mana, N.; Pianesi, F.; Chippendale, P.; Lanz, O.; and Potamianos, G. 2009. Multimodal classification of activities of daily living inside smart homes. In Omatu, S.; Rocha, M. P.; Bravo, J.; Fern'andez, F.; Corchado, E.; Bustillo, A.; and Corchado, J. M., eds., *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living*, 687–694. Berlin, Heidelberg: Springer Berlin Heidelberg.

Monekosso, D.; Florez-Revuelta, F.; and Remagnino, P.

2015. Ambient assisted living [guest editors' introduction]. *IEEE Intelligent Systems* 30(4):2–6.

Numao, M., and Masuda, S. 2016. Multimodal classification of activities of daily living inside smart homes. In *Non-Restrictive Continuous Health Monitoring by Integration of RFID and Microwave Sensor*.

Pryor L., C. G. 1996. Planning for contingencies: a decision-based approach. *Journal of Artificial Intelligence Research 4* 287–339.

Reviews, C. 2016. *Artificial Intelligence*. Cram101.

Settles, B. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Estimation of Personalized Value through the Analysis of Conversational Data Assisted by Coimagination Method

Mihoko Otake,[1] Masato S. Abe,[1] Masahiro Nochi,[2] Eiji Shimizu[3]

1. RIKEN Center for Advanced Intelligence Project
2. Graduate School of Education, University of Tokyo
3. Graduate School of Medicine, Chiba University
{mihoko.otake, masato.abe} 'at' riken.jp (replace 'at' to @)

Abstract

People refer to personalized value to actively make decision on his/her own life. It is important to understand how personalized value is developed in adolescence and how it influences later life. Methods for assisting development of personalized value may contribute to human wellbeing. In this study, we report the estimated personalized value in adolescence through the analysis of conversational data assisted by coimagination method, which has been used for older adults. We found that the method is effective for estimating attitude towards everyday life, when the theme of conversation was "favorite snacks", which may help introspecting the personalized value of each participant.

Introduction

Personalized value is a person's inner drive for long-term action, which will be internalized and personalized through adolescence. Adolescence is characterized by social interactions with peers while childhood is associated with transgenerational incorporation of parental values. It is thought that personalized value may help to pursue subjective well-being (Fukuda and Kasai 2017).

Linguistic and psychological intervention methods such as cognitive behavioral therapy may have preventive and therapeutic effects on mental illness. It has been shown that cognitive therapy is as efficacious as antidepressant medications at treating depression (DeRubeis et al. 2008). Yoshinaga et al. (2016) revealed that clinical trials have clarified that cognitive behavioral therapy is effective for patients with social anxiety who do not improve with antidepressants. Although cognitive behavioral therapy has mainly involved interventions by language, methods using images are also attracting attention (Blackwell et al. 2015). Likewise, the assistive technology to develop personalized value may contribute to human happiness. Estimation method for identifying personalized value is one of the fundamentals to achieve the goal because feedback of the estimated value may help introspecting the personalized value of each person.

Otake et al. (2011) proposed conversational assistive technology named coimagination method where themes, allocated period for speech, listening, questions and answers are predetermined so that all participants can participate in the conversation in an equal manner. Participants take photos beforehand so as to show them during the periods of speech, questions and answers. Participants can share their episodes with photos and co-imagine them with each other. It applied mainly to older adults and the basic effect and safety were confirmed (Otake et al. 2013).

We have preliminary determined that selected topics according to the theme reflect the personalized value of each speaker. In this study, we report the estimated personalized value of adolescence through the analysis of conversational data assisted by coimagination method.

Method

We conducted group conversation sessions supported by coimagination method. In this study, the theme of the conversation session was "favorite snacks". All 12 participants (9 women, 3 men) were high school first grade students and were divided into 3 groups by 4 people. In the group conversations, each participant had a talk about her/his favorite snack (1 minute) and conducted Q & A session (2 minutes) while displaying photos of favorite snacks collected in advance. Then, we analyzed the transcribed data of group conversation and searched for items of contents that are considered to reflect the personalized value of each participant.

Results

We extracted items that are associated to the personalized values from the topics provided by the participants. It was found that topics provided by the "favorite snacks" are classified by the attitude toward ordinary or extraordinary life scenes of each participant. Many participants refer to the price (cheap or expensive), the amount of eating (a large or a few), the frequency of eating (often or rarely), and the tension between their appetite and body shape.

Table 1 shows the number of topics classified in terms of price, amount, and frequency in all sessions. Favor on ordinary or extraordinary life scenes are classified based on either of the price, amount and frequency of eating of the snacks. If the snack is cheap, eaten a large amount, often, then the participant who provided the topic may favor ordinary life scene. In contrast, if the snack is expensive, eaten a few amount, rarely, then the participant may favor extraordinary life scene.

We found that 8 participants favored ordinariness, 3 participants favored extraordinariness, 1 participant favored both (Table 1). Overall, there were more participants who wanted to eat a large amount of favorite snacks or eat frequently, namely, emphasized ordinariness. Participants who chose extraordinariness, such as rarely eating, small amount of the snacks, are supposed not to be able to eat them by external factors such as high price and their body shape, and without those factors it is thought that they could eat more. Interestingly, the topic of roll cake was classified in both. This reflects that it can be enjoyed both in daily snack scenes and extraordinary scenes such as parties and Christmas. It was explicitly stated that the participant put value on both ordinariness and extraordinariness.

Table 1. Number of topics in all sessions

	price	amount	freq.	ordinary
cheap/many/often/ ordinary	5	5	4	8
expensive/a few /rare/extraordinary	2	0	3	3
both	1	0	0	1
total	8	5	7	12

Discussion

We extracted items related to personalized values from topics of 12 participants. In all generations, it may be possible to classify the conversation into ordinariness and extraordinariness according to the items extracted in this research, in the coimagination method which is carried out by the theme of "favorite snacks" or "favorite food". This can reveal their personalized values.

There are some participants who favored extraordinariness. However, whether they also favor ordinary snacks or only favor extraordinary slacks is unknown because the participants were allowed to select only one favorite slack during the experiment. In order to make clear this condition, the number of topics should be more than one so that the participants can select multiple favorite things.

Conclusion

In this study, we presented the estimated personalized value in adolescence through the analysis of conversational data assisted by coimagination method. We found that the method is effective for estimating attitude towards everyday life, when the theme of conversation was "favorite snacks", which may help introspecting the personalized value of each participant. Future work includes comparison of measured personalized values based on personalized value questionnaires and estimated ones from the conversational data, evaluating whether feedback of the estimated value may help introspecting the personalized value of each person. We will explore how taking photographs based on a theme, searching for topics, and sharing them in group conversation can lead to assist personalized value development in adolescence.

Acknowledgement

This study was supported by JSPS KAKENHI Grant Number JP16H06395 and 17H05920.

References

Blackwell, S. E. et al. 2015. Positive Imagery-Based Cognitive Bias Modification as a Web-Based Treatment Tool for Depressed Adults: A Randomized Controlled Trial, Clin Psychol Sci. 3(1):91-111.

DeRubeis, R. J. et al. 2008. Cognitive therapy vs. medications for depression: Treatment outcomes and neural mechanisms, Nat Rev Neurosci. 9(10): 788-796.

Kasai, K., Fukuda, M. 2017. Science of recovery in schizophrenia research: brain and psychological substrates of personalized value. *npj Schizophrenia* 3:14.

Otake, M., Nergui, M., Otani, T., and Ota, J. 2013. Duplication Analysis of Conversation and its Application to Cognitive Training of Older Adults in Care Facilities, Journal of Medical Imaging and Health Informatics. 3(4): 615 - 621.

Otake, M., Kato, M., Takagi, T., & Asama, H. 2011. The Coimagination Method and its Evaluation via the Conversation Interactivity Measuring Method, Early Detection and Rehabilitation Technologies for Dementia: Neuroscience and Biomedical Applications, Jinglong Wu (Ed.), IGI Global: 356 - 364.

Yoshinaga, N., et al. 2016. Cognitive behavioral therapy for patients with social anxiety disorder who remain symptomatic following antidepressant treatment: a randomized, assessor-blinded, controlled trial. Psychotherapy and psychosomatics. 85(4): 208-217.

Hybrid Sensing and Wearable Smart Device for Health Monitoring and Medication: Opportunities and Challenges

Mahboob Qaosar, Saleh Ahmed, Chen Li, Yasuhiko Morimoto
Graduate School of Engineering, Hiroshima University
Kagamiyama 1-7-1, Higashi-Hiroshima 739-8521, Japan
Email: {d172517, d162694, d165000, morimo}@hiroshima-u.ac.jp

Abstract

Global health-care systems are struggling with rapid increasing of aging population, the prevalence of chronic diseases, and raising of medical treatment costs. In this paper, we proposed a hybrid sensing and wearable device for health informatics and emergency medication. The proposed device will include some of the existing individual modules for monitoring health attributes and emergency medication. Moreover, it will also include information communication modules, which will assist the prescribed physician and health center to monitor the patient remotely. In addition, the communication modules will enable the device to communicate automatically with emergency medical services when needed. Furthermore, the proposed device will also act as a virtual medical assistant to advice regular medicine to the patient according to his/her prescription.

1 Introduction

Global health-care systems are struggling with rapid increasing of aging population, the prevalence of chronic diseases, and raising of medical treatment costs (Bloom et al. 2011). Now it becomes the major challenge of our aging society. It is also found that elderly people tend to be very unconscious about their health condition and regular medical check-up.

With rapid increase of aging people with chronic diseases, we need to find smarter ways to manage the health needs without increase of financial burden of hospitals and/or nursing facilities. To address these incomplete health-care requirements, particularly for the early diagnosis and treatment of major diseases, remote biomedical sensing device has raised as an active area of interdisciplinary research. According to (Zheng et al. 2014), a significant progress in developing health-monitoring systems for health-care applications have been made in the past decade, but most of them are still in their prototype stages. There are some major challenges like user acceptability, portability, reduction of motion artifact, power consumption, self-processing capability, and distributed interference in wireless communication networks still need to be considered to increase the usability and functions of these devices for practical use. Also, there are several biomedical instruments have already developed for emergency treatment, which can save the human life in some unpredictable circumstances.

Taking all these issues in mind, we are proposing a hybrid sensing and wearable device for health monitoring and emergency medication. The proposed hybrid-device will include some of the existing individual modules for monitoring blood glucose, blood-pressure, heart rate, body temperature, and ECG. On the other hand, it will also facilitate some treatments and medications for emergency situation like automatic drug injection to human body. As a medical assistant, it can also suggest regular routine medicine to the patient based on physician's prescription and also advise emergency drug/medicine to the patient depending on the health monitoring attributes. It should include facilities to communicate with the registered health center for automatic interchange of regular health-informatics, prescription, and emergency services.

The proposed device will help aging people especially those who are suffering from chronic diseases like diabetes, high blood pressure, cardiac arrhythmia, and cardiovascular diseases. Such kind of instrument will open up a new dimension in biomedical instrumentation and artificial intelligence research areas. Moreover, the data acquired from the individual sensor will be a large and valuable source of the knowledge storage, which will play an important role in the development and expansion of medical science. Such kind of device will introduce new challenges to the researchers too. For example, accuracy improvement of the health monitoring attributes, miniaturization and unobtrusiveness of the wearable device, reliability of data-communication interface, on-node intelligent data processing and power consumption.

In this paper, we review some modules and technologies from the published scientific research papers in Section 2, which could be integrated with the proposed device for sensing and processing physiological data. In Section 3, we discuss the main objectives and characteristics of the proposed device. The significance, opportunities, and effectiveness of the proposed device are discussed in Section 4. Then we show the major challenges in Section 5, which should be needed to overcome for the development and implementation of such device. Finally we conclude the paper in Section 6.

2 Related Works

Because of some features of unobtrusive and wearable devices have transformed their usages for biomedical measurement devices, since they have been widely used in the clinical environment for many decades due to the recent advances in sensing and networking technologies. Fortunately, with the rapid progress of integrated circuit technologies and micro-electromechanical technologies, the size of processing electronics and measurement modules have been significantly shrunk for wearable and portable applications.

A compact micro-system could be deployed for monitoring cardiac electrical and mechanical activity, which may combine the multi-sensor modules, signal processing unit, and battery unit into a single integrated platform. Similarly, a pressure-free and cuff-less measurement module is desirable to monitor patient's arterial blood pressure (BP) continuously. In this regard, the pulse transit time (PTT) based cuff-less blood pressure measuring technology could be chosen considering the comforts of user by replacing the widely used cuff-based methods, as proposed in (Poon and Zhang 2005). However, the PTT based cuff-less blood pressure meter is still in experimental stage.

Again, most common ways to check glucose levels involves pricking a fingertip with a lancing device to obtain a blood sample. And then using a glucose meter to measure the blood sample's glucose level, which will not be suitable for monitoring blood glucose continuously or periodically. In this regard, continuous glucose monitoring (CGM) systems, using a tiny sensor inserted under the skin to check glucose levels in tissue fluid, would be a better solution for the proposed device as described by (Ahmadi and Jullien 2009). Besides that, it may also integrate a minimally invasive and pseudo-continuous blood glucose monitoring system as described in (Wang et al. 2017), which extracts a whole blood sample from a small lanced skin wound using a shape memory alloy (SMA)-based micro-actuator for measuring the blood glucose level directly from the sample.

As for ECG, although the capacitive coupling sensing is the most commonly used technology for capturing ECG signals, it is not suitable for perpetual monitoring, since this technology requires contact gel, which provides direct resistive contact to the subject body (Zheng et al. 2014). Garment-integrated sensing is another technology for continuously monitoring physiological parameters. The garment-integrated sensing active electrode was presented in some papers for wearable ECG monitoring (Pani et al. 2016; Nemati, Deen, and Mondal 2012; Fuhrhop, Lamparth, and Heuer 2009; Yama, Ueno, and Uchikawa 2007; Park et al. 2006), which could be introduced for the proposed device.

Self-assessment during vital condition of the patient required technologies like micro-controller based electromechanical injection syringe pump proposed in different papers (Koundinya et al. 2014; Chirgwin, LaCourse, and McWilliam 2015; Rideout et al. 2011; Chee, Fernando, and Trinh 2006). These could be utilized for injecting life saving drug(s) to a cardiac patient or for injecting regular insulin to diabetic patients. It could be also utilized for injecting glucose to a diabetic patient in case of hypoglycemia symptom.

Furthermore, several papers was published and proposed different methods for assisting a patient with chronic diseases. An interactive robot for reminding medication has been proposed by (Datta et al. 2012). A personalized diet and exercise guideline recommendation system was proposed by (Tseng et al. 2015). Personalized medical assistant base preventive health-care model was also proposed by (Aridarma, Mengko, and Soegijoko 2011). On the other hand, the largest Information Technology(IT) companies like Google, Apple, Microsoft, Amazon and Facebook has already introduced their virtual personal assistant *Google Assistant*, *Siri*, *Cortana*, *Alexa* and *M*. So, the biomedical instrument developer industries may collaborate with these IT companies to integrate the conversational interactive voice response interface and the artificial intelligence(AI) technologies provided by these virtual personal assistant, to implement Virtual Personal Medical Assistant(VPMA).

Finally, networking will be an integral part of proposed wearable devices to deliver high-efficiency and high-quality health-care services to support the remote health monitoring. Body Sensor Network (BSN) is presently a very popular research topic and extensive progresses have been made in the past decades. (Yadav and Tripathi 2017) has proposed adaptive clustering scheme for effective data communication in health-care monitoring system. Self-adaptive data collection and fusion for health monitoring has beet proposed by (Habib et al. 2016). On the other hand, event-driven middleware besed on smartphone has been proposed in (Seeger, Laerhoven, and Buchmann 2015). Recently, scientists have drawn significant research attention on introducing Internet of Things (IoT) in the health-care system, which integrates the Internet with remote monitoring smart sensors and medical device. Many articles were published on IoT-based applications for health-care system (Nguyen et al. 2017; Span, Pascoli, and Iannaccone 2016; J and Shivashankar 2017). Security and reliability of the communication media would be the major focus point in this regard. Several articles has been published, surveyed and proposed techniques for securing the communication media for BSN (Naik and Samundiswary 2016; Chukwunonyerem, Aibinu, and Onwuka 2014).

3 Objectives

Basically, acquisition of health-related information by unobtrusive sensing and wearable technologies is considered as a cornerstone in health informatics (Zheng et al. 2014). Health informatics mainly deals with the acquisition, transmission, processing, storage, retrieval, and use of different types of health and biomedical information (Zhang and Poon 2010). The two main acquisition technologies of the health attributes are sensing and imaging. But we are preferring only the sensor technologies, specially the unobtrusive sensing and wearable modules for continuous health monitoring. In order to acquire health information continuously and pervasively in daily living, sensors can be integrated with garments or wearable accessories. The sensors can also be organized as stick-on electronic tattoos. Moreover, for enabling long-term health monitoring, it can be printed onto the human skin.

The main objective of unobtrusive health sensing is to

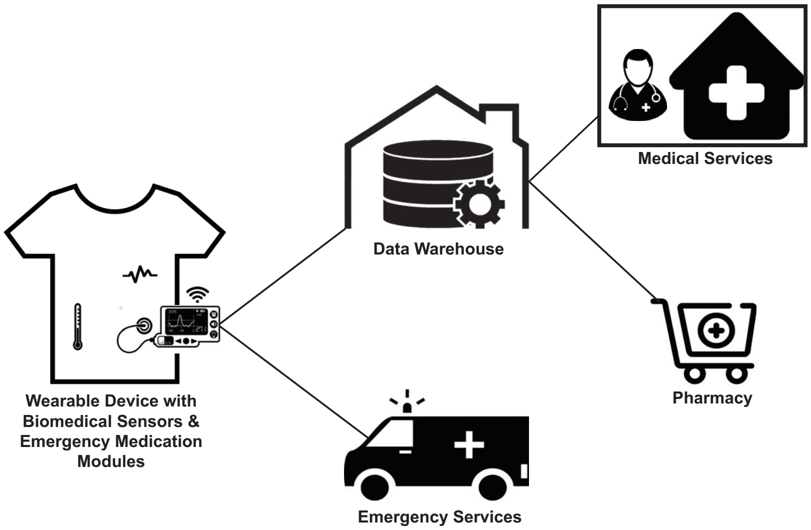

Medical Services
Data Warehouse
Wearable Device with Biomedical Sensors & Emergency Medication Modules
Pharmacy
Emergency Services

Figure 1: System diagram of the proposed wearable smart device for health monitoring and medication

enable continuous monitoring of physical activities and behaviors, as well as physiological and biochemical parameters during the patient's daily living. The most commonly measured vital signs include: Electrocardiography (ECG), blood-glucose, heart rate, blood pressure (BP) and surface body temperature. Different data acquisition modules, described in Section 2, can provide real-time information and facilitate timely remote intervention to acute events such as high blood glucose, hypoglycemia, high blood pressure, cardiovascular diseases, and cardiac arrhythmia, particularly in rural and remote areas where expert treatment is not available. Therefore, considering all these health factors and recent progress in health technologies, we are proposing to conduct a research on designing and developing a compact wearable device, which will integrate all the existing modules to monitor most commonly measured vital signs without users initialization. It will also concentrate on sending the acquired personalized bio-medical attributes' records to the health center in discrete and customizable time intervals by the widely available Internet infrastructure, using cellular or Wi-Fi wireless communication media.

The proposed device should also be capable of taking necessary initiatives for emergency medication and treatment in case of unpredictable situation and notifying the patient about regular medication routine like a medical assistant according to physician prescription. It may also notify the prescribed physician about current health condition of the patient.

Miniaturization and unobtrusiveness should need to be carefully considered so that the proposed device is comfortable for users. Further more, the device may also act as an intelligent agent terminal for patient, which will interact with physician and pharmacy on behalf of the patient. Where the physician may prescribe medicine to the patient based on health informatics provided by the sensors of the proposed device and then the agent may track the physician's prescription to place orders automatically to the pharmacy. After receiving orders, pharmacy may deliver the medicine to corresponding patient and also notify the agent about orders status. Then, the present stock of individual medicine, further consultation requirement with the physician and new orders to the pharmacy may also be tracked by the agent.

Figure 1 describes the conceptual system diagram of the proposed wearable smart biomedical device. Here, the device will interchange information directly with the data warehouse. Only the prescribed physicians and health center can see the individual's biomedical attributes from the data warehouse. In case of life-threatening situation, the proposed device will directly communicate with emergency medical services automatically.

As we have discussed in Section 2 of this paper, a significant progresses in developing required modules for the proposed device have made in the past decades. Although most of them are still in their prototype state, the proposed device does not seems to be unrealistic. It should be required to choose right modules for integration that will serve demand. Further research would be conduct to refine those selected modules for improving the accuracy, efficiency, unobtrusiveness and power consumption.

4 Significance

A significant portion of medical expenditure in the world is spent on managing chronic disease in the hospital. Since the average life expectancy of the people is increasing all over the world (Wikipedia), the demand for the hospitals, aged care services, nurses, and doctors are also increasing. The proposed device will be suitable for remote health monitoring in such areas, which has a significant number of elderly citizens; since with the increase in the age of people different chronic diseases and their symptoms are begin to be observed in human body, such as diabetes, high blood pressure, cardiovascular diseases, and cardiac arrhythmia. Moreover, for elderly people, the automated smart monitoring system can provide detailed information about the continuous changes in their health conditions, which will help the well-wishers keep their intensive observation. It will promote the healthy lifestyle. It can also detect health risk and facilitate the implementation of preventive measures at an earlier stage. In addition, examining the person's health condition from the everyday life is more effective than the clinical setup.

Besides that, regular monitoring of the patient's physical condition will help the patient for follow-up discussion with the doctor. It will also help to alert physician about the vital changes in patient's health condition and facilitate patient self-assessment of chronic disease.

The physiological situation of a patient with chronic diseases is most commonly managed by using vital signs monitoring of health attributes; which includes asthma, diabetes, hypoglycemia, high blood pressure, cardiovascular diseases and cardiac arrhythmia. Monitoring these vital signs help the patient and caretaker to coordinate treatment and medication dosage (e.g. insulin in the case of diabetes, anti-hypertensive medication in the case of high blood pressure).

Finally, it will reduce the risk of patient's life, especially those who live alone, since it will alert the specific authorities automatically in the patient's risky circumstances without human interaction. It will also provide emergency medication and treatment service to the patient. As explained earlier, the senior citizens are not sincere about their health condition, it becomes difficult to monitor their health condition regularly. And when they live alone in rural places, then it becomes more difficult to provide emergency services, since in some vital life-threatening situation, like hypoglycemia, high blood pressure or heart diseases, the patients lost their sense. And at that time they become unable to communicate with the nearest health center or their well-wishers for the emergency support by their self. Remote monitoring and emergency medication may help us to deal with these vital signs, which is particularly well suited to Africa, Australia and Asia, which have significant of the population living in rural and remote areas.

Moreover, continuous health-monitoring systems have been shown to be effective in helping to manage chronic disease, post-acute care, and monitoring the safety of the aging people. Regular health-monitoring technologies can help elderly people to slow the progression of chronic disease and can ensure continued recovery after being discharged from an acute care setting. Unobtrusive and automatic health monitoring technologies can also alert caregivers and prompt intervention when a vulnerable old person is injured or in harms way.

Beside that, the monitoring device will be useful for rhythm monitoring to understand the cardiac role of unexplained symptoms. Furthermore, it would also support arrhythmia medication therapy to monitor treatment effectiveness.

The structured data acquired from the individual devices would also be a key source for data-mining. The data resource can be used for clinical decision support systems. It can be applied for the up-gradation of proposed device itself. The proposed device will be cost efficient because it may achieve considerable cost savings in a number of aspects such as reducing visits to specialists. It avoids symptom exacerbations that lead to hospitalizations, which reduces potentially preventable hospitalizations (PPHs), age related PPHs, emergency room visits, and nursing home admissions. It also keeps low-care residential patients in their homes, which decreases the burden on health care professionals, patient transport costs, and hospitalizations.

5 Challenges

As discussed in Section 2 most of the proposed modules are in prototype stage which cannot be applicable in real world applications immediately. Considering as a biomedical instrument the proposed smart device should need to be highly sophisticated and accurate, as well as be intelligent enough to initiate emergency services in a vital condition. So, we cannot proceed with those modules and methods, until they are proven to be accurate and capable of dealing with different circumstances. Therefore, a vast study should be required to choose each individual module for monitoring the health attributes and medication, considering their availability, accuracy, performance, miniaturization, unobtrusiveness and data-communication interface.

On the other hand, energy efficiency will also be a crucial element for the proposed wearable device since it will directly affects the design and usability of the device, especially for long-term monitoring applications. This motivate us to consider the power consumption as a vital indicator of performance evaluation to select the modules for the proposed device.

Another challenging task will be the development of the embedded software for the device. It should need to be intelligent and perfect to take an emergency decision in a life-threatening health state without communicating with any Internet-based knowledge. The software will directly deal with automatic initiation of each individual sensing components for new reading with customizable time interval. It will also deal with post processing of sensed data and local memory management system. In addition, it will manage regular data exchange between the device and the data warehouses and will also manage communication with the emergency service provider when needed. Finally, the software should be focused on the optimization of applications that will not only improve the performance of the proposed device but also ensure the best utilization of hardware resources and power.

As a virtual personal medical assistant, the device should be interactive to the users by a friendly interface. Some virtual personal assistant applications, such as Apple Siri, Google Assistant, Microsoft Cortana, can make the device more interesting to the users. For example, the proposed device may remind the patient verbally for regular routine medication. It may also warn the patient after detecting any dangerous sign in the sensed biomedical attributes.

Finally, as a wearable biomedical instrument software, it should be not only focus on the improvement of performance, but also ensure the maximal utilization of hardware resources and power consumption. In this context, the development of proposed device requires collaboration between industries and researchers, who are developing the technologies for the biomedical instrument.

6 Conclusion

In this paper, an overview of hybrid health monitoring and medication platforms in wearable form is presented. In Section 2, we have shown that the underlying technology pave the way for implementation of proposed device. We have also shown the significance and the major challenges for implementing such devices. Since the researchers and industries are continuously inventing and producing new technologies and gadgets for human welfare, we can say that in near future we may observe this kind of devices. The future development of the proposed device will greatly rely on the advances in a number of different areas such as materials, sensing, energy efficiency, electronics and information technologies for data acquisition, transmission and analysis due to the multidisciplinary nature of this research topic. The proposed device will be cost-effective because of it's high volume of demands. So, we propose collaboration between industries, researchers and developers to share their technologies and innovative ideas to joint together for developing such kind of devices with health monitoring and medication capabilities. It will make the architecture of these device unique and make the platform open to all industries for production of such devices. It is also required to enforce identical data structure for information interchange. We believe that with the future implementation of the proposed device will advance the medical services and lead to fundamental changes in how the health-care service will be delivered in the future.

7 Acknowledgments

This work is supported by KAKENHI (16K00155,17H01823) Japan. Mahboob Qaosar and Saleh Ahmed are supported by Japanese Government MEXT Scholarship.

References

Ahmadi, M. M., and Jullien, G. A. 2009. A wireless-implantable microsystem for continuous blood glucose monitoring. *IEEE Transactions on Biomedical Circuits and Systems* 3(3):169–180.

Aridarma, A.; Mengko, T. L.; and Soegijoko, S. 2011. Personal medical assistant: Future exploration. In *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, 1–6.

Bloom, D. E.; Cafiero, E. T.; Jane-Llopis, E.; Abrahams-Gassel, S.; Bloom, L. R.; Fathima, S.; Feigl, A. B.; Gaziano, T.; Mowafi, M.; Pandya, A.; Prettner, K.; Rosenberg, L.; Seligman, B.; Stein, A. Z.; and Weinstein, C. 2011. The global economic burden of non-communicable diseases. *Geneva, Switzerland: World Economic Forum*.

Chee, F.; Fernando, T. L.; and Trinh, H. M. 2006. Microprocessor-based insulin delivery device with amperometric glucose sensing. In *First International Conference on Industrial and Information Systems*, 65–68.

Chirgwin, C. W.; LaCourse, J. R.; and McWilliam, P. 2015. Smart syringe: Determining the depth and location of the needle during intramuscular injection. In *2015 41st Annual Northeast Biomedical Engineering Conference (NEBEC)*, 1–2.

Chukwunonyerem, J.; Aibinu, A. M.; and Onwuka, E. N. 2014. Review on security of wireless body area sensor network. In *2014 11th International Conference on Electronics, Computer and Computation (ICECCO)*, 1–10.

Datta, C.; Tiwari, P.; Yang, H. Y.; Kuo, I. H.; Broadbent, E.; and MacDonald, B. A. 2012. An interactive robot for reminding medication to older people. In *2012 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 190–190.

Fuhrhop, S.; Lamparth, S.; and Heuer, S. 2009. A textile integrated long-term ecg monitor with capacitively coupled electrodes. In *2009 IEEE Biomedical Circuits and Systems Conference*, 21–24.

Habib, C.; Makhoul, A.; Darazi, R.; and Salim, C. 2016. Self-adaptive data collection and fusion for health monitoring based on body sensor networks. *IEEE Transactions on Industrial Informatics* 12(6):2342–2352.

J, J. J., and Shivashankar. 2017. A survey on wireless body sensor network routing protocol classification. In *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, 489–494.

Koundinya, P.; Mallikarjuna, M.; Vinod, S.; Devaraj, D.; and Reddy, B. M. K. 2014. Reconfigurable hardware for patient monitoring systems. In *International Conference on Information Communication and Embedded Systems (ICICES2014)*, 1–6.

Naik, M. R. K., and Samundiswary, P. 2016. Wireless body area network security issues - survey. In *2016 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT)*, 190–194.

Nemati, E.; Deen, M. J.; and Mondal, T. 2012. A wireless wearable ecg sensor for long-term applications. *IEEE Communications Magazine* 50(1):36–43.

Nguyen, H. H.; Mirza, F.; Naeem, M. A.; and Nguyen, M. 2017. A review on iot healthcare monitoring applications and a vision for transforming sensor data into real-time clinical feedback. In *2017 IEEE 21st International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, 257–262.

Pani, D.; Dess, A.; Saenz-Cogollo, J. F.; Barabino, G.; Fraboni, B.; and Bonfiglio, A. 2016. Fully textile, pedot:pss based electrodes for wearable ecg monitoring systems. *IEEE Transactions on Biomedical Engineering* 63(3):540–549.

Park, C.; Chou, P. H.; Bai, Y.; Matthews, R.; and Hibbs, A. 2006. An ultra-wearable, wireless, low power ecg monitoring system. In *2006 IEEE Biomedical Circuits and Systems Conference*, 241–244.

Poon, C. C. Y., and Zhang, Y. T. 2005. Cuff-less and non-invasive measurements of arterial blood pressure by pulse transit time. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, 5877–5880.

Rideout, T. M.; LaCourse, J. R.; McWilliam, P. L.; and Evans, E. J. 2011. Force sensing syringe to analyze needle path forces during intramuscular injection. In *2011 IEEE 37th Annual Northeast Bioengineering Conference (NEBEC)*, 1–2.

Seeger, C.; Laerhoven, K. V.; and Buchmann, A. 2015. Myhealthassistant: An event-driven middleware for multiple medical applications on a smartphone-mediated body sensor network. *IEEE Journal of Biomedical and Health Informatics* 19(2):752–760.

Span, E.; Pascoli, S. D.; and Iannaccone, G. 2016. Low-power wearable ecg monitoring system for multiple-patient remote monitoring. *IEEE Sensors Journal* 16(13):5452–5462.

Tseng, J. C. C.; Lin, B. H.; Lin, Y. F.; Tseng, V. S.; Day, M. L.; Wang, S. C.; Lo, K. R.; and Yang, Y. C. 2015. An interactive healthcare system with personalized diet and exercise guideline recommendation. In *2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 525–532.

Wang, G.; Poscente, M. D.; Park, S. S.; Andrews, C. N.; Yadid-Pecht, O.; and Mintchev, M. P. 2017. Wearable microsystem for minimally invasive, pseudo-continuous blood glucose monitoring: The e-mosquito. *IEEE Transactions on Biomedical Circuits and Systems* 11(5):979–987.

Wikipedia. Life expectancy.

Yadav, D., and Tripathi, A. 2017. Load balancing and position based adaptive clustering scheme for effective data communication in wban healthcare monitoring systems. In *2017 11th International Conference on Intelligent Systems and Control (ISCO)*, 302–305.

Yama, Y.; Ueno, A.; and Uchikawa, Y. 2007. Development of a wireless capacitive sensor for ambulatory ecg monitoring over clothes. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 5727–5730.

Zhang, Y. T., and Poon, C. C. Y. 2010. Editorial note on the processing, storage, transmission, acquisition, and retrieval (p-star) of bio, medical, and health information. *IEEE Transactions on Information Technology in Biomedicine* 14(4):895–896.

Zheng, Y. L.; Ding, X. R.; Poon, C. C. Y.; Lo, B. P. L.; Zhang, H.; Zhou, X. L.; Yang, G. Z.; Zhao, N.; and Zhang, Y. T. 2014. Unobtrusive sensing and wearable devices for health informatics. *IEEE Transactions on Biomedical Engineering* 61(5):1538–1554.

From Algorithms to Heuristics: Will Androids Ever Make Freudian Slips?

Sadeq Rahimi, PhD

In-Sync Strategy / Harvard Medical School / Boston Graduate School of Psychoanalysis
Sadeq.rahimi@insyncstrategy.com / Sadeq_rahimi@hms.harvard.edu

Abstract

This paper presents a conceptual framework for understanding the importance and relevance of human cognitive biases in development of more effective and seamless human-AI interactivity. A categorization model is proposed for distinguishing two general types of human biases: 1) biases that are primarily determined by neurobiological hardwiring, and 2) those that are determined and transmitted across generations by cultural and symbolic structures of thought and affect. It will then examine the meaning and implications of the suggested types of biases, specifically in relation to human-AI interaction.

One of the most intriguing aspects of writing this paper for me has been the wonderfully paradoxical fact that it has compelled me to look simultaneously back and forward in time: back to a paper I wrote long ago in defense of the notion of "cultural logic"; and forward, of course, towards the concept of artificially intelligent agents demonstrating a degree of subjectivity. It would perhaps make good logical sense for me to start with the earlier place and then move to the one ahead.

Reasoning, I had argued in an earlier discussion of cultural logic (Rahimi, 2002), is a process that occurs within the space of meanings defined by linguistic, social, and cultural environments; and logic, or the process of "establishing necessary connections between these meanings", as the great theorist of structuralism, Levi Strauss, once put it (Levi Strauss, 1966, p. 35), is always already defined by and bound within these linguistic, social and cultural parameters. Consider the following excerpt, which is drawn from an interview reported by Michael Cole, a psychologist who studied the relationship between culture, intelligence and logical reasoning. This is how a conversation between the interviewer and a Kpelle tribal elder in Liberia unfolded (Cole & Scribner 1974, p. 162):

I: […] spider and black deer always eat together. Spider is eating. Is black deer eating?
E: Were they in the bush?
I: Yes.
E: Were they eating together?
I: Spider and black deer always eat together. Spider is eating. Is black deer eating?
E: But I was not there. How can I answer such a question?
I: Can't you answer it? Even if you weren't there, you can answer it [repeats the question].
E: Oh, oh, black deer is eating.
I: Why?
E: The reason is that black deer always walks about all day eating leaves in the bush. Then he rests for a while and gets up again to eat.

I should mention that this same excerpt is used in Cognitive Psychology textbooks to demonstrate what we call "belief bias," where incorrect conclusions are assumed to be valid simply because they are consistent with personal beliefs (see *e.g.* Kellogg, 2016, p. 321). I will not enter a discussion of the specific cognitive processes involved here, but I mentioned this example primarily as an instance of the ways in which culture, cultural norms and culture-specific models of interpreting events and the environment directly influence reasoning, giving rise to what we can call a "cultural logic" various aspects of which may or may not coincide with the model of reasoning employed in a different culture. To quote Cole himself, "the notion of culture-free intelligence is a contradiction in terms" (Cole, 1999, p. 645).

Before coming back to make more sense of this brief introduction, let us examine the specific sense in which cognitive biases are of interest to us here, and the implications of that for thinking about biases in relation to AI.

It is a truism of the day to speak of big data, deep learning and artificial intelligence as parts and parcels of the same assemblage. As clearly as deep learning and hence AI are currently dependent on big data, however, there is no good reason to expect such dependency to be permanent. Intelligent machine's dependence on man-made sets of data, no matter how extensive or how intricate those sets might become, is perhaps one of the last obstacles on its way to achieving fully integrated general AI. I am not an AI expert, but even from where I stand as a social scientist, in the imaginable ideal future of artificial intelligence the machine will be able to connect, interact with, collect, and process 'real world' information without the need for human intermediation. Given this distinction, it is possible, perhaps necessary, to consider two paradigmatically distinct fields of contemplation on the questions of cognitive processes, specifically cognitive biases, in relation to AI. The first would concern the long-term developmental framework of artificial intelligence and artificial subjectivity, and the second would concern current and near-future relational framework of human-AI interaction.

Whereas in the context of the long-term developmental trajectory of artificial intelligence the main question would concern the "subjectivity" of artificially intelligent agents as such, the primary concern of the shorter-term human-AI relation, which defines the scope of the current paper, should be human subjectivity, and the ability of AI to serve and interact with human subjects in a fashion most desired by and beneficial to the latter.

In order to briefly consider the point of view in which cognitive biases are understood as a fundamental point of distinction between human subjectivity and artificial intelligence, let us start with a basic distinction between the "thinking" and "decision making" processes in the two. Let us consider the simple fact that, while AI is programmed to pursue a line of rational thinking fed with specific sets of data and driven by specific algorithms to fulfill specific objectives, humans' reality consists of formative real-life uncertainties and is driven by the mechanics of cognitive heuristics (biases) and emotional forces (desires). Indeed these two distinctly implicit and distinctly human forces are both defined in contradistinction to the basic notion of rationality, and research tells us, both of which are highly informed by environmental, specifically social and cultural parameters. As this audience knows better than most, human cognitive processes, specifically cognitive biases which serve as the disciplinary hallmark of Behavioral Economics, are gradually emerging as significant differentiators between human and artificial intelligence. They are typically introduced in precisely the frame that we are considering here, namely the distinction between the natural and embodied workings of *homo sapiens'* reasoning and decision making processes, versus the "machine like" reasoning paths and decision making steps taken by *homo economicus* (Thaler, 2000), which represents a conceptual, abstracted and disembodied entity.

Foregoing once again a vast and vastly tempting discussions of behavioral economics and the characteristics that set homo sapiens apart from homo economicus, I would like to focus here on cognitive biases as a phenomenological category, specifically in its relevance to thinking about AI. As you know, a long list of human psychological processes have been generally grouped by cognitive scientists, and hence within the language of Behavioral Economics, under the basic descriptor of "biases." However, almost all common lists of cognitive biases in fact contain a range of different biases that we need to conceptualize in a more specific way, if we are to make more precise use of them beyond the basic sense that 'biases are processes that warp people's perceptions and hence affect their decision making.' This fact has been driven home for me especially strongly in the context of market related social and psychological research, where the focus is often on development and design of appropriate, specific, and effective strategic interventions (aka *nudges*. See Thaler & Sundstein, 2008).

Generally speaking, the so-called "cognitive biases" on most given lists can be understood as occupying different places on a bi-polar spectrum that ranges from neurobiological processes to symbolic processes. The two poles of this scale are therefore populated by what we can call neuro-perceptual biases on one end and socio-cultural biases on the other. And somewhere between these two poles we can also identify primarily cognitive or primarily emotive biases, though this latter distinction will have to remain primarily conceptual, since in reality the cognitive and the emotive are often too intricately entangled to comfortably tease apart. We can think for instance of a group of biases that are identified primarily with cognitive processes such as decision paralysis, metaphorical thinking, or dissonance reduction bias; versus those associated strongly by emotional forces, such as selective attention, ostrich effect, or own-group bias. It is important to keep in mind, however, that there is a great degree of melding and overlap between these two types, since in reality most biases are driven in varying proportions by both cognitive and emotional factors. Let us pause briefly to consider neuro-perceptual and socio-cultural biases, before moving on to examine the significance of such classification of biases.

Neuro-perceptual biases can be understood as a group of biases that originate from the interaction of our neurological system and the sensory inputs, such as sight, touch, smell, taste, or hearing. These biases function as lenses that impact our direct impressions of the environment, even before such impressions have been cognitively interpreted or made symbolic or linguistic sense of. Basic visual or auditory biases belong to this group. Think, for instance, of studies that have uncovered biases in visual and auditory localizations: people are biased towards the center when localizing visual stimuli, and biased towards the periphery when localizing auditory stimuli (Odegaard et al., 2015). Our sensory perception of the world around us is in fact biased by an interaction of neurological processes and prior expectations, rather than directly reflecting the actual qualities of our environment. As early as the 19th century, Helmholtz (1867) used the term "unconscious inference" in describing the process through which sensory data is mixed with prior knowledge and existing expectations to create a perception of the environment. More recently, Patten et al. (2017) investigated biases in the visual processing of spatial orientation and they demonstrate how prior expectations and current sensory information interact to generate inescapable biases in visual perception.

Socio-cultural biases, on the other hand, are rooted in culturally endorsed patterns of thought and interpretation, including such phenomena as cultural logic, and they lead to culturally specific emotional and cognitive modes of sense making and symbolic interpretation of objects, events or actions. A growing body of research is showing us that culturally-driven cognitive and emotive biases play as much an impact on individual decision making as do neurologically-driven biases. Researchers such as Miamoto (2002, 2013) or Kitayama and Park (2016) have highlighted the presence of clear links between local cultural patterns of thought and prominence of certain types of cognitive biases over others. For instance, research has repeatedly demonstrated significant cross-cultural differences in presence of dispositional bias (aka the fundamental attribution error), which is the idea that in interpreting and judging other people's behavior, we tend to assign a heavy weight to internal (dispositional) factors such as their character and their intention, rather than circumstantial causes of the behavior, while we tend to do the reverse when judging our own (Morris and Peng, 1994; Chua et al., 2005; Kitayama et al., 2006; Kitayama et al., 2009; Choi et al., 1999; Miyamoto and Kitayama, 2002). In addition to patterns of thought, culturally endorsed social biases also contribute to formation of individual biases. These are cultural biases that lead to interpreting and judging phenomena by standards that are inherent to one's own culture. Consider for example the case of a teacher who may have a favorite student, driven by the infamous own-group favoritism bias. The bias can lead to teacher's ignoring that student's sub-optimal performance, while s/he might not notice the overachievement of another student whom s/he perceives less favorably due to same culturally endorsed biases. The following example, which I am quoting from a clinician, offers a good example of cultural bias:

I'm treating a Native American patient. When I ask questions, she consistently avoids meeting my eyes. I interpret this as evasiveness, shyness, and lack of assertiveness. As a result, I arrive at the incorrect interpretation that she is currently being abused because she's acting so submissive. In reality, according to her cultural blueprints, averting her eyes is a sign of respect, which she is trying to afford me as her physician.

As mentioned earlier, the "socio-cultural" aspects of such biases can be traced back to a number of factors, such as cultural logic, local semiotic and linguistic processes, local cultural norms and behavioral blueprints, and culture-specific emotional patterns, which has also been called 'structures of feeling' (Sharma & Tygstrup, 2015; Williams, 1977). Earlier I briefly mentioned the notion of cultural logic. Metaphors, which are the bread-and-butter of human communication, are developed and used in different ways and to different extents across cultures; leading to significant divergences in modalities of cognitive and symbolic processing of information, as well as behavioral and decision making patterns across cultures. A powerful body of research has driven the point home that metaphors represent local cultural conceptual systems of thinking, and can be studied as such (e.g. Lakoff & Johnson, 1980; I-wenSu, 2002; Zhou, 2009). In my own work on culture and subjectivity I have provided extensive analyses of culture-specific patterns of metaphoric referencing that govern patterns of thought, emotion and decision making (e.g. Rahimi, 2015, 2016). Think of the color red, for instance, and the divergent implications it may have across cultures such as in Turkey (where red is the primary color of the flag and tied to strong nationalist sentiments); the United States, the USSR, or China.

Culture regulates, structures and provides guidelines and expectations for understanding, experiencing, and expressing emotions. In addition to distinct cross-cultural differences in context-dependent expression and interpretation of emotions, cultural differences exist also in evaluation and social consequences, of emotions (Jenkins & Karno, 1992; Scherer, 2000). As demonstrated by psychological anthropologists, Jean Briggs, in her famous book *Never in Anger* (Briggs, 1970), the Utku Eskimo culture enforced strict limits on feeling and expressing anger, so that anger was rarely communicated, and in the rare occasions where it did occur, it was reacted to by such strict measures as ostracizing the individual. Psychologists Ekman and Friesen (1975) addressed the ways culture's "unwritten codes" become internalized by individuals growing into a society, and how the internalized codes continue to structure the ways in which emotions are felt and expressed by those individuals.

To recap the above in terms relevant to the question of human-AI relations, two main groups of biases can be distinguished. One group includes biases that are embedded in and transmitted across human generations via the symbolic structures that underlie collective systems of meaning and power. These are biases that I have grouped under socio-cultural biases. These are embedded implicitly in such processes as local systems of meaning, local metaphoric patterns and what we term 'cultural logic.' A second group of biases, however, may be identified as those that are based on non-algorithmic shortcuts known as cognitive heuristics. While the first set of biases are embedded and transmitted through the collective level of structures and relations of meaning, the second is embedded within the individual/private realm of cognitive mechanisms and psychological dynamics, and primarily driven through universal neurological features of our species.

Insofar as the question of AI-human interactivity is concerned, the two group of biases may call for different approaches and concerns. The socio-cultural group of biases, for instance, may consist primarily of biases that can be detected and, depending on specific objective, either prevented or leveraged, through systematic examination of data collection and management, as well as in programming of our learning and intelligence agents. Anybody following the AI news has no doubt noticed the recent outpouring of articles and research on such topics as "racist AI" or "sexist AI". The same biases that are "transmitted" implicitly across generations within a given culture through such media as linguistic and semiotic processes are obviously bound to be "transmitted" also to AI agents through the vast amount of data that is created within the same social and cultural systems of meaning and points of reference. In order to better understand, and therefore detect and either prevent or perhaps leverage such biases, we will need to turn our focus on the "data" as such: we will need to try rigorously to identify semiotic, logical, and emotive patterns that function as the warp and weft of our social and cultural fabric and our collective systems of meaning.

With the second group of biases, those driven primarily by neuro-cognitive processes and which function at the level of the individual, however, we may notice the need for an altogether different approach. This set of biases will not be transmitted to our machines through terabytes of data, and they will likely remain to represent a significant gap between human and machine cognitive processes. This is a problem solving which may in fact require an unusual step of intentionally introducing cognitive biases that would warp the machine's cognition or solutions and decisions in a specific direction, in order to either make the human-AI interactions more coherent and "natural"; or make AI more powerful in understanding and anticipating "human" patterns of thought, emotion or behavior and decision making. We may, in other words, have no option but to intentionally incorporate or represent cognitive biases into the workings of our otherwise un-biased intelligent agents. In addition to rendering intelligent agents more compatible with human cognitive and emotional needs and nuances, such cognitive warp filters may in fact be necessary in the context of research, specifically health related research, where we would need AI to identify such deeply human, and deeply biased behaviors as treatment adherence (or lack thereof), and the range of cognitive biases associated with it.

References

Briggs, J. L. (1970). *Never in anger: Portrait of an Eskimo family* (Vol. 12). Harvard University Press.

Choi, I., Nisbett, R. E., & Norenzayan, A. (1999). Causal attribution across cultures: Variation and universality. *Psychological Bulletin*, *125*, 47-63.

Chua, H.F., Leu, J., Nisbett, R.E. (2005b). Culture and diverging views of social events. Personality and Social Psychology Bulletin, 31, 925–934.

Cole, M. (1999). Culture free Versus Culture Based Measures of Cognition. In Sternberg, RJ (Ed) *The Nature of Cognition*, pp. 645-664. Cambridge, MA: MIT press.

Cole, Michael; Scribner, S. 1974. *Culture and Thought: A Psychological Introduction*.

New York: Wiley.

Ekman, P., & Friesen, W. V. (1975). Unmasking the face: A guide to recognizing emotions from facial cues.

Helmholtz, H.v (1867) *Treatise on physiological optics, Vol III*. Trans. and ed. JPC Southall. (Translated from the 3rd German edition, English edition 1962) Dover, UK.

I-wen Su, L. (2002). What Can Metaphors Tell Us About Culture? *Language and Linguistics*, 3(3), 589-613.

Jenkins, J. H., & Karno, M. (1992). The meaning of expressed emotion: Theoretical issues raised by cross-cultural research. *The American Journal of Psychiatry*, *149*(1), 9.

Kellogg, R. T. (2016). *Fundamentals of Cognitive Psychology*. London: Sage.

Kitayama, S., Park, J. (2010). Cultural neuroscience of the self: understanding the social grounding of the brain. *Social cognitive and affective neuroscience*, *5*(2-3), 111-129.

Kitayama, S., Park, H., Sevincer, A.T., Karasawa, M., Uskul, A.K. (2009). A cultural task analysis of implicit independence: comparing North America, Western Europe, and East Asia. Journal of Personality and Social Psychology, 97, 236–255.

Kitayama, S., Mesquita, B., Karasawa, M. (2006). The emotional basis of independent and interdependent selves: Socially disengaging and enga- ging emotions in the US and Japan. Journal of Personality and Social Psychology, 91, 890–903.

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by*. University of Chicago press.

Lévi-Strauss, Claude. 1966. *The Savage Mind*. Chicago: University of Chicago Press.

Miyamoto, Y., & Wilken, B. (2013). Cultural Differences and Their Mechanisms. In D. Reisberg (Ed.), *The Oxford handbook of cognitive psychology*, pp. 970-985. Oxford: Oxford University Press.

Miyamoto, Y., & Kitayama, S. (2002). Cultural variation in correspondence bias: the critical role of attitude diagnosticity of socially constrained behavior. *Journal of personality and social psychology*, *83*(5), 1239.

Morris, M.W., Peng, K. (1994). Culture and cause: American and Chinese attributions for social and physical events. Journal of Personality and Social Psychology, 67, 949–971.

Odegaard, B., Wozny, D. R., & Shams, L. (2015). Biases in visual, auditory, and audiovisual perception of space. *PLoS computational biology*, *11*(12), e1004649.

Patten, M. L., Mannion, D. J., & Clifford, C. W. (2017). Correlates of perceptual orientation biases in human primary visual cortex. *Journal of Neuroscience*, *37*(18), 4744-4750.

Rahimi, S. (2002). Is cultural logic an appropriate concept? A semiotic perspective on the study of culture and logic. *Sign Systems Studies*, *30*(2), 455-467.

Rahimi, S. (2015). *Meaning, madness and political subjectivity: A study of schizophrenia and culture in Turkey*. London: Routledge.

Rahimi, S. (2016). Haunted metaphor, transmitted affect: The pantemporality of subjective experience. *Subjectivity*, *9*(1), 83-105.

Scherer, K. R. (2000). A cross-cultural investigation of emotion inferences from voice and speech: Implications for speech technology. In *Sixth International Conference on Spoken Language Processing*.

Sharma, D., & Tygstrup, F. (Eds.). (2015). *Structures of feeling: Affectivity and the study of culture* (Vol. 5). Walter de Gruyter GmbH & Co KG.

Thaler, R. H. (2000). From homo economicus to homo sapiens. *Journal of economic perspectives*, *14*(1), 133-141.

Thaler, R., Sunstein, C. (2008). Nudge: The gentle power of choice architecture. *New Haven, Conn.: Yale*.

Williams, R. (1977). Structures of feeling. *Marxism and literature*, *1*, 128-35.

Zhou, D. (2009). *Dynamics in Metaphor Comprehension-A Cross-cultural Web-based Experiment on Understanding Teacher Metaphors* (Doctoral dissertation, Universität Duisburg-Essen, Fakultät für Ingenieurwissenschaften)

Robinson, A. L. 1980a. New Ways to Make Microcircuits Smaller. *Science* 208:1019-1026.

Sleep Stage Re-Estimation Method According to Sleep Cycle Change

Yusuke Tajima,[†] Akinori Murata,[†] Tomohiro Harada,[††] Keiki Takadama[†]
[†]The University of Electro-Communications, Chofugaoka 1-5-1, Chofu-shi, Tokyo-to, Tokyo, 182-8585 Japan
[††]Ritsumeikan University, Nojihigashi 1-1-1, Kusatsu-shi, Shiga-ken, 525-8577 Japan
[†]{y_tajima, kouho.aki}@cas.lab.uec.ac.jp, [††]harada@ci.ritsumei.ac.jp, [†]keiki@inf.uec.ac.jp

Abstract

This paper focuses on a sleep cycle, and improves the problem which an estimation accuracy of Real-Time Sleep Stage Estimation Method(RSSE) when it estimates a sleep stage on real time. Concretely, the proposed method re-estimates the sleep stage immediately after first sleep cycle since going to bed for the problem which decreases the correct rate of the sleep stage estimated by RSSE as time passes since going to bed. From the human subject experiments, the following implications have been revealed: (1) the correct rate improved by re-estimation in 8 cases out of 9 cases. (2) when the sleep cycle is long, it is possible to calculate the sleep cycle from the same subject's past sleeping information and if it is used, the estimation accuracy is improved for all cases.

Introduction

Currently, it is said that 20% of people are sleeping disorder patients in Japan, and it is considered to be a large proportion. Sleep disorders are classified finely according to the symptoms, and there are many symptoms from mild such as poor sleep to serious such as narcolepsy. As a cause of the increase in the number of patients with sleep disorders, modern social life can be considered. For example, the opportunity to experience jet lag is increasing due to the development of traveling means, and sleeping time is reduced by working time system regardless day and night such as night shift. Also, the quality of sleep reduces due to mental damage due to stress. In order to treat these sleep disorders, it is necessary to observe the sleep stage which is degree of sleep state in addition to the effect of medication. Even in terms of prevention against sleeping disorders, that is, maintaining the quality of sleep with a high level, the importance of recording daily sleep states is high. The deterioration of the quality of sleep also exists as a sign of diseases such as mental illness, so to maintain human health care the observation of sleep state is importance. In addition, there is a great correlation between the quality of sleep and the quality of work, so a bad sleep is feared that economic losses such as traffic accidents. Furthermore, in recent years, there is a mindfulness as an action to raise the productivity of daytime activities, it is thought that it can raise higher productivity by controlling sleeping.

How do you record the sleep stage? The Rechtschaffen & Kales method (R&K method) exists as a method most used as a method of acquiring the sleep state. This method acquires the electroencephalogram (EEG), the electromyogram (EMG), and the electrooculogram (EOG) data by attaching multiple electrodes to the subject's head and face. From these acquired data, specialist doctors classify sleeping state into six states or four states. The thing obtained in this way is the sleep stage and it makes it possible to know objectively the depth of sleep. Generally, it is said that the sleep stage acquired by the R&K method shows the sleep state with 80% accuracy. Therefore, the specialist doctor does not diagnose and classify by one self, but multiple doctors diagnose and classify for maintaining high accuracy. However, the following two problems exist in the R&K method: (1) it takes time to acquire the sleep stage because it is necessary to classify it by diagnosis of a specialist doctor, (2) this method requires the connection of many devices (including many electrodes) to the human body, increasing the stress on the human subjects. Because of these problems, the R&K method is not a practical method in terms of recording the daily sleep state.

In order to solve the problem of (1), there is a neural network which uses a learn data as obtained from the electrodes and uses a teacher data as the sleep stage diagnosed by specialist doctor. However, this method purpose is support to inexperienced doctor of the sleep diagnosis, not for sleep state recording. On the other hand, solving the problem (2) is indispensable for recording to sleep state. In order to solve the problem of (2), a method that does not require attachment of electrodes, diagnosis and classification without specialist doctors for the same sleep stage obtained by the R&K method is being studied. Concretely, Heart rate, body movement, respiration, and body temperature exist as biological data that can be acquired instead of EEG, EMG, and EOG, which is data obtained by attaching electrodes. Watanabe acquired heart rate, body movement, and respiration by analyzing the frequency of data obtained from a pressure sensor placed under the bed. In addition, there are methods using head movement and thermography during sleeping, in this thesis from the viewpoint of ease of introduction of instruments, biological data (heart rate, body movement) obtained from a pressure sensor is used. Many researches have been done on methods for estimating the sleep stage without using expert knowledge from acquired biological data. Watanabe estimate the sleep stage by extracting the medium frequency component from the obtained heartrate and discretizing it. This uses knowledge that the wave shape of the medium frequency component of the heartrate correlates with the sleep stage. Based on the findings used by Watanabe, Harada determined frequency bands related to the sleep stage beforehand and used trigonometric function approximation for them so that the sleep stage could be estimated in real time. In this study, we are looking for an action that leads to good sleep during sleep (for example, shed sound leading to sleep) to improve future insomnia and prevent it, so we will focus on the estimation Harada's method. Harada's estimation method has a problem that the coincidence ratio (hereinafter referred to as ”accuracy rate”) with the sleep stage derived by the R&K method after sleep deteriorates. Therefore, this research address this problem by using the sleep cycle. Specifically, since the sleep cycle that appears for the first time after going to bed is different from other sleep cycles among one sleep (indicating from sleeping to getting up), it is estimated again at the end of the first sleep period. As the judgment of the end of the sleep cycle, REM sleep estimated by Harada's method is used. The effect of this estimation method will be clarified by subject experiment.

The rest of this paper is organized as follows. First, the previous work related to the sleep stage estimation is introduced in Section 2. Section 3 describes the proposed method that estimates the sleep stage again based on the sleep cycle. Section 4 describes the experiments conducted on the subjects and presents the obtained results. Finally, the conclusions of this paper are presented in the final section.

Previous Method

Rechtschaffen & Kales Method

The Rechtschaffen and Kales method: R&K method is the gold standard method to determine the sleep stage and is treated generally for a treatment of sleep disorders. Concretely, the medical experts determine the sleep state which is classified into the six stages from a viewpoint of depth of sleep estimated by the data of electroencephalogram (EEG), the electromyogram (EMG), and the electrooculography (EOG). Since the accuracy of the sleep stage in this method is enough for a medical purpose, this method has been widely employed as the global standard method. But this method requires to connect many devices (including many electrodes and codes) to human body, which increases the stress of human subjects. Since this reason, this method is not suitable as determining the daily sleep stage for health care.

Real-Time Sleep Stage Estimation Method

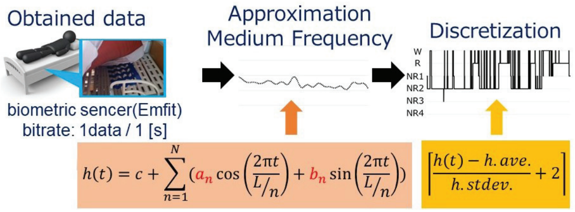

Obtained data
biometric sencer(Emfit)
bitrate: 1data / 1 [s]
Approximation
Medium Frequency
Discretization
W
R
NR1
NR2
NR3
NR4
$h(t) = c + \sum_{n=1}^{N}(a_n \cos\left(\frac{2\pi t}{L/n}\right) + b_n \sin\left(\frac{2\pi t}{L/n}\right))$
$\left\lceil \frac{h(t) - h.ave.}{h.stdev.} + 2 \right\rceil$

Figure 1: Overall flow of RSSE

In order to estimate the sleep stage from heartrate in real-time, Harada proposed the sleep stage estimation method: RSSE using trigonometric function approximation. Figure 1 shows the overall flow of the RSSE. This method starts to approximate the heartrate as the trigonometric function and estimates the sleep stage from the intermediate frequency range of the approximate heartrate. The approximate heartrate is modeled as follows.

$$h(t) = c + \sum_{n=1}^{N}(a_n cos(\frac{2\pi t}{L/n}) + b_n sin(\frac{2\pi t}{L/n})) \qquad (1)$$

In this equation, ($h(t)$ denotes the estimated heartrate at time t, L denotes the maximum period of the intermediate frequency component(in this method L uses the value 2^{14}), N denotes the number of composed trigonometric functions(in this method N uses the value 14). The model parameters an, bn and c are calculated by the maximum likelihood estimation method that minimizes the following likelihood function.

$$minJ = \frac{1}{T}\sum_{t=1}T(HR(t) - h(t))^2 + \frac{\lambda}{N}\sum_{n=1}^{N}(a_n^2 + b_n^2) \quad (2)$$

RSSE is able to estimate the sleep stage without connecting biometric sensor and expert knowledge. But RSSE has the problem which cannot accurately estimate the sleep stage when a fixed time has passed since going to bed.

Proposed Method

We expect that the problem of RSSE is change of sleep cycles. Sleep cycles are the period from REM sleep to REM sleep. In Figure 2, this sleep has 4 sleep cycles. The front 3 cycles have 60 minutes cycle, but the back cycle has 80 minutes cycle. This difference influence the accuracy rate of RSSE. Therefore, this study proposes the Re-Estimation method which restart RSSE at the point where changing the sleep cycles. In particular, RSSE is restarted before changing sleep cycle, in Figure 2, the restarted point is end of the 60 minutes cycles. In the proposed method, the change in sleep cycle is judged by focusing on REM sleep. Because the sleep cycle refers to the interval from REM sleep to REM sleep, the appearance of REM sleep represents a change in sleep cycle. Among them, since sleep first sleep cycle has quite different nature in other sleep cycle, the proposed method re-estimates when REM sleep appears for the first time in sleep stage estimation by RSSE. However, since WAKE and REM sleep are not stable immediately after going to bed, the proposed method is supposed to re-estimate when the REM appears for the first time after 45 minutes or more has passed since going to bed. This is due to the fact that the time of sleep cycle is about 60 to 120 minutes if there are individual differences.

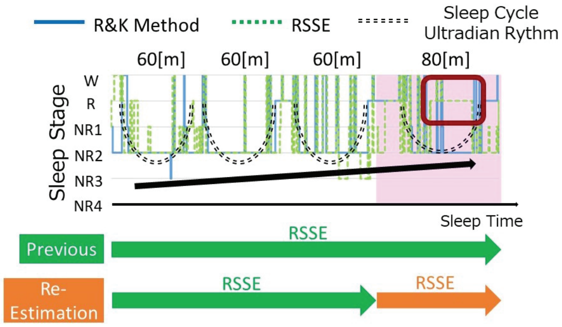


Figure 2: Approach to change sleep cycle

Table 1: Details of subjects

	age	Sex
Subject A	20	Male
Subject B	40	Male
Subject C	60	Female

Experiment

To investigate the effectiveness of the Re-Estimation method, we conducted the human subject experiment as the field experiment. In particular, we compared the RSSE(All Time) as the previous method and RSSE(Re-Estimation) which re-estimated at the position which is changed the sleep cycle.

Setup

To investigate the effectiveness of the proposed method, we conducted human subject experiment. Table 1 shows the details of the three healthy subjects with no sleeping disorders. The sleep for the three days of each subject was measured. Two types of measuring instruments were used for measuring sleep in this experiment. One is AlicePDx in Fig.- which is a kind of the electroencephalograph used to measure the subjects electroencephalogram(EEG), electromyogram(EMG) and electrooculogram(EOG), the other is Emfit in Fig.- which is a non-connect biosensor used to measure the subjects heartrate, body movement and respiration. In this experiment, the subject attached some electrodes for measuring by AlicePDx, and slept on the bed where Emfit was lain. The data measured by AlicePDx (EEG, EMG and EOG) is used to calculate the sleep stage by the R&K method, whereas the data measured by Emfit (heartrate, body movement and respiration) is used to estimate the sleep stage by the real-time sleep stage estimation method. In the R&K method, medical specialists determine the sleep stage every second of sleep, and in the real-time sleep stage estimation, the sleep stage is estimated using data measured at every second by Emfit. On the day of the experiment, we asked subjects to refrain from excessive exercise and from drinking alcohol. In addition, there were no external factors that prevent sleeping such as alarms, and subjects were able to fall asleep and get up whenever they want. The parameters used for the estimation of the sleep stage in Real-time sleep stage estimation method are listed in Table 2.

Table 2: Parameters in RSSE

Parameter	Value
L	$2^{14} \approx 4.5\text{h}$
λ	1
t_{int}	60s=1min
N	13

Evaluation Criteria

The sleep stage which is estimated by the re-estimation method compared with the sleep stage is estimated by the R&K method. Because the R&K method is the gold standard method all over the world, the sleep stage estimated by this method is treat correct. In this experiment, the sleep stage which is derived by re-estimation method compares the correct sleep stage. In comparison, REM sleep 1 and REM sleep 2, REM sleep 3 and REM sleep 4 are combined, in other word, this experiment evaluates in 4 stages instead of 6 stages. Then the purpose of this study is to increase the correct rate of sleep stage estimation when sleep cycle is changed. Calculation of the accuracy rate is calculated from the time when 30 minutes have elapsed from the re-estimation time to the time to get up. This is because RSSE has a problem that adaptation occurs immediately after estimation and estimate worse accuracy. In order to prevent this, it is suggested to use the past sleep data, but since there is no clear means yet. So this experiment excluded the time from calcurating the accuracy rate.

Result

Table 3: Experiment Sleep Data

Subject	Day	Bedtime	Wake-up	Re-Estimation
SubjectA	1 day	23:18:40	06:29:08	03:05:00
	2 day	02:18:19	07:02:17	03:23:00
	3 day	01:28:58	09:08:46	05:08:00
SubjectB	1 day	00:07:56	04:57:36	02:26:00
	2 day	01:50:07	05:47:43	02:33:00
	3 day	01:30:09	05:42:25	02:16:00
SubjectC	1 day	22:26:34	06:43:46	02:46:00
	2 day	00:20:44	06:10:32	04:30:00
	3 day	22:06:46	06:02:42	02:27:00

Table 3 shows sleep information of Subjects(Bedtime, wakeup time, and re-estimation time) used in actual experiments. Figure 3 is the result of this experiment. In this figure, the vertical axes shows accuracy rate%, the horizontal axes shows sleep data of three subjects. The horizontal stripes pattern shows RSSE(All Time) result, the fill point pattern shows RSSE(Re-Estimation). In this figure, the correct rate of RSSE(Re-Estimation) results except for subjectB day3 are higher than the correct rate of RSSE(All Time). The maximum value of the improvement ratio of the correct rate is 15% at subjectA Day1.

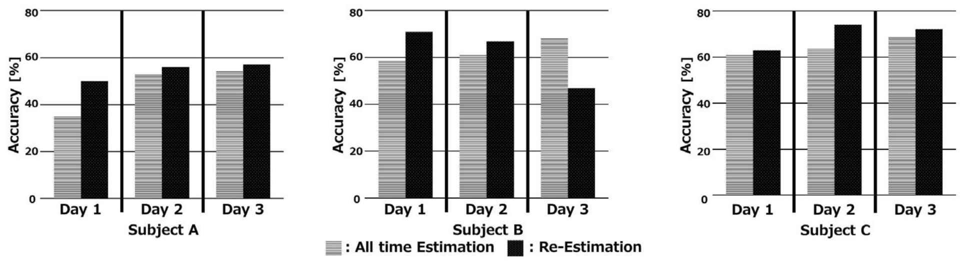


Figure 3: Experiment result

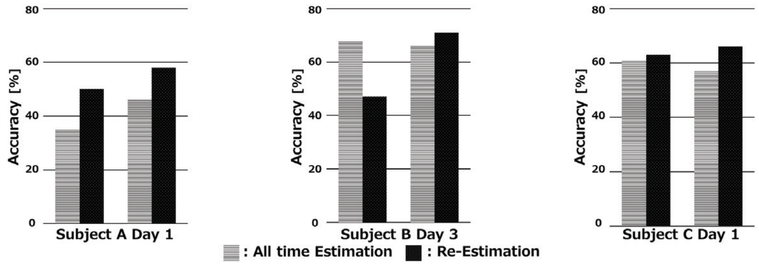


Figure 4: Estimated result after improvement

In addition, Figure 4 shows the sleep stage diagrams of each subject in four stages. The vertical axis shows the sleep stage (Wake, REM, Shallow REM, Deep REM), and the horizontal axis shows sleeping time. The sleep stage map for one day is one set with two figures, the upper shows R&K method and RSSE (All-Time), the lower shows R&K method and RSSE (Re-Estimation). The solid lines in the figure show the sleep stage diagram of the R&K method respectively, the gray dotted line is RSSE (All-Time), and the gray double-dotted line is RSSE (Re-Estimation). In each figure, the place where the background is gray shows the part used for calculating the coincidence rate. From the figure, as in Day 2 of subject B, the accuracy of sleep cycles was properly captured by re-estimating the first sleep cycle in some cases, but as in Day 1 of subject C, the time to re-estimation was long There are cases in which it takes time. In the case of the latter case, even though the accuracy is improved, it may be thought that the first sleep cycle to be captured has not been captured inherently.

Discussion

When the RSSE REM sleep judgment does not appear properly, like Day1 of SubjectA, Day3 of SubjectB,and Day1 of SubjectC, a problem arises in which re-estimation does not occur for a long time. This is because the REM judgment of RSSE will not be exactly the same as the R&K method. In order to solve this problem, only when the REM sleep does not appear for a certain period of time, the proposed method re-estimated based on the time of the sleep cycle inherent to the individual. Specifically, we use the time of the sleep cycle that first appeared by looking at the past sleep. In this experiment, the sleep cycle time of subject A was 70 minutes, subject B was 70 minutes, subject C was 100 minutes. If the REM sleep does not appear even after this time, the result of performing the re-estimation is shown in Fig4.

Conclusion

This paper focused on the sleep stage estimation method based on the sleep cycle change, and improved its estimation accuracy by re-estimation. More specifically, it is a method of capturing REM sleep first appeared in RSSE as a change in sleep cycle and re-estimating RSSE again. Using the human subject experiments, the following conclusions have been revealed: (1) the correct rate improved by re-estimation in 8 cases out of 9 cases. (2) when the sleep cycle is long, it is possible to calculate the sleep cycle from the same subject's past sleeping information and if it is used, the estimation accuracy is improved for all cases.

It should be noted here that the results have been obtained from only three subjects, which means that further careful qualifications and justifications, such as an increase of the number of subjects, are necessary to generalize our results. In addition to this important direction, the following issues must be addressed in the near future: (1) since sleep is affected by internal and external factors such as the subject's health condition, previous night's sleep conditions, daytime activities, weekly variation, and seasonal effects, these should be addressed in the next stage of this research; (2) since the accuracy of our proposed method differs with the estimation time from 50% to 90%, we should improve our method for more stable and accurate sleep stage estimation by addressing,(i) the use of the body movement and respiration data for the estimation of the sleep stage.

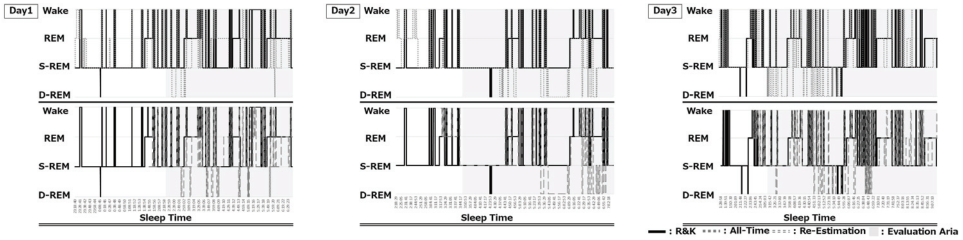


Figure 5: Sleep Stage of SubjectA

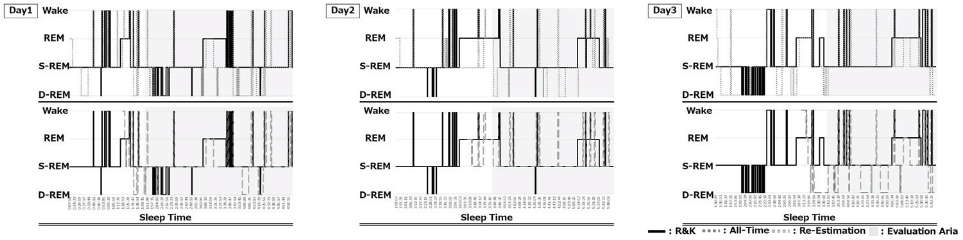


Figure 6: Sleep Stage of SubjectB

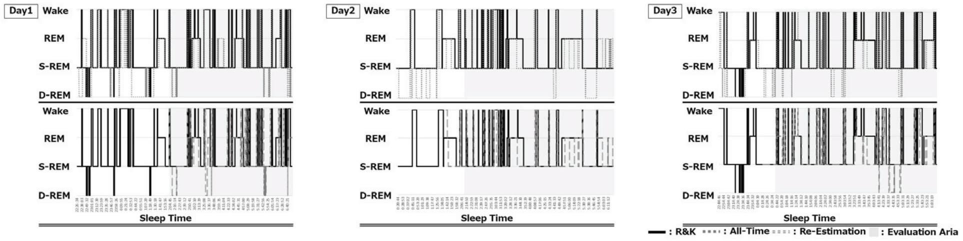


Figure 7: Sleep Stage of SubjectC

References

Rechtschaffen, A., & Kales, A. *A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects.*, Bethesda, Md., U. S. National Institute of Neurological Diseases and Blindness, Neurological Information Network, 1968

Harada, T., Uwano, F., Komine, T., Tajima, Y., Kawashima, T., Morishima, M., & Takadama, K.: Real-Time Sleep Stage Estimation from Biological Data with Trigonometric Function Regression Model, *AAAI Spring Symposium Series*, 2016.

Takadama, K., Hirose, K., Matsushima, H., Hattori, K., and Nakajima, N.:"Learning Multiple Band-Pass Filters for Sleep Stage Estimation: Towards Care Support for Aged Persons,"*IEICE (The Institute of Electronics, Information, and Communication Engineers) Transactions on Communications*, Vol. E93-B, No. 04, 811/818 (2010)

Watanabe, T. and Watanabe, K.: Noncontact Method for Sleep Stage Estimation,*IEEE Transaction on Biomedical Engineering*, 10-51, 1735/1748 (2004)

Shimohira, M.; Shiiki, T.; Sugimoto, J.; Ohsawa, Y.; Fukumizu, M.; Hasegawa, T.; Iwakawa, Y.; Nomura, Y.; and Segawa, M. Video analysis of gross body movements during sleep, *Psychiatry Clin Neurosci* 52-2, 176/177 (1998)

Harper, R. M.; Schechtman, V. L.; and Kluge, K. A. Machine classification of infant sleep state using cardiorespiratory measures. *Electroencephalography and Clinical Neurophysiology*, 67-4 379/387 (1987)

Can Machine Learning Correct Commonly Accepted Knowledge and Provide Understandable Knowledge in Care Support Domain? Tackling Cognitive Bias and Humanity from Machine Learning Perspective

Keiki Takadama

The University of Electro-Communications, Japan
keiki@inf.uec.ac.jp

Abstract

This paper focuses on care support knowledge (especially focuses on the sleep related knowledge) and tackles its *cognitive bias* and *humanity* aspects from machine learning perspective through discussion of whether machine learning can correct commonly accepted knowledge and provide understandable knowledge in care support domain. For this purpose, this paper starts by introducing our data mining method (based on association rule learning) that can provide only necessary number of understandable knowledge without probabilities even if its accuracy slightly becomes worse, and shows its effectiveness in care plans support systems for aged persons as one of healthcare systems. The experimental result indicates that (1) our method can extract a few simple knowledge as understandable knowledge that clarifies what kinds of activities (*e.g.*, rehabilitation, bathing) in care house contribute to having a deep sleep, but (2) the *apriori* algorithm as one of major association rule learning methods is hard to provide such knowledge because it needs calculate all combinations of activities executed by aged persons.

1. Introduction

Most of commonly accepted healthcare-related knowledge are subjective or exaggerated from a small number of evidence. An example includes the knowledge that "the eight hours are needed for good sleep." This knowledge is somewhat correct and wrong because the deepness of sleep is more important than the length of sleep from the viewpoint of good sleep. This kind of knowledge tends to be commonly accepted because of an effect of the *focusing illusion* (Kahneman 2006) as one of *cognitive bias* (Kahneman 1982, Gilovich, 2002, Haselton 2005) that occurs when people place too much importance on one aspect of an event. In detail, the focusing illusion is caused by a wrong impression due to a strong focus of a specific aspect of an event (*e.g.*, the length of sleep) as the expected outcome (*e.g.*, good sleep), which may derive the wrong outcome (*e.g.*, better sleep is derived by a longer sleep). Such an expected but wrong outcome is called as *illusion* in the context of the focusing illusion. For more its understanding, let me show the following typical question: "Do people become happier when they were richer?" This question focuses on an amount of money as the only evaluation criterion of happiness. Needless to say, however, our happiness is not only determined by an amount of money but also by other evaluation criteria such as good wellness, and good relationship of family.

To overcome such cognitive bias, *machine learning* (Mitchell 1997) has a great potential of correcting commonly accepted knowledge because it can provide the different knowledge extracted in a rational manner. In particular, big healthcare data recently becomes personal, which enables machine learning to extract the personalized knowledge from such data. This promotes us to place an importance of the personalized knowledge in comparison with the commonly accepted knowledge.

What should be noted here, however, is that most of machine learning methods are hard to provide the understandable knowledge to human. For example, *deep learning* algorithm (Hinton 2006a, Hinton 2006b, LeCun 2015) has made revolution in computer vision, speech recognition, and natural language processing, but we cannot know how deep learning algorithm learns. In other words, deep learning algorithm contributes to increasing the accuracy of predictions, but it cannot provide the knowledge that we can understand. This indicates that we can only operate deep learning algorithm as a black box tool because the network structure acquired by deep learning algorithm is very complex. From this fact, we need a new machine learning that can provide the understandable knowledge to human, *i.e.*, the knowledge with a high human readability. For this issue, *association rule learning* (or *rule-based machine learning*) such as apriori algorithm (Agrawal 1993, Agrawal 1994) is appropriate because it can discover interesting relations among variables (inputs) in large databases as the understandable knowledge.

However, the apriori algorithm needs to calculate all combinations of variables as rules, which results in providing a lot of knowledge to human. Furthermore, such knowledge has the following two probabilities: (i) *confidence* (which is an indication of how often the rule has been found to be true) and (ii) *support* which is an indication of how frequently the rule appears in the dataset. Considering these features of the apriori algorithm, we do not understand all combinations of variables as rules but select the distinctive combinations for easy understanding. Even after such a selection, we do not still understand the rules *with* some probabilities, but prefer to understand the rules *without* probabilities. For example, we do not prefer the rule such as "exercise is good for deep sleep" with 70% probability under 10% appearance in the dataset, but prefer the only rule without probabilities even though such a rule is not 100% correct. This indicates that humans prefer to understand something as simply as possible even in a little ambiguous understanding. If this kind of understanding is based on *humanity*, what we need in healthcare domain is a new machine learning based on humanity, *i.e.*, the method that can provide only necessary number of understandable knowledge without probabilities even if its accuracy slightly becomes worse. For this purpose, this paper starts by introducing our rule-based machine learning method and shows its effectiveness in *care support systems* for aged persons as one of healthcare systems. Concretely, this paper focuses on care support knowledge (especially focuses on the sleep related knowledge) and tries to extract a few significant understandable knowledge that clarifies what kinds of activities (*e.g.*, rehabilitation, bathing) in care house contribute to having a deep sleep

This paper is organized as follows. The next section explains an overview of machine learning and Section 3 introduces the care plan and daily activities scheduled in the care plan. Section 4 shows our previous human subject experiments. Finally, the conclusion is given in Section 5.

2. Machine Learning

Classification of machine learning

Machine learning is roughly categorized as follows.

- **Supervised learning**
  The goal of this learning is to learn the relationship between inputs and their desired outputs provided by a "teacher". Concretely, this learning aims at generating the function ($y=f(x)$) that maps inputs (x) to outputs (y). Note that this *function* can be called as a *model* which is represented by neural network (Rumelhart 1986) model, rule-based model including tree structure, Bayesian model, and so on. Major algorithms in supervised learning include (1) *classification* (which aims at assigning unseen inputs to one or more of these classes); and (2) *regression* (which aims at estimating output value from inputs).
- **Unsupervised learning**
  The goal of this learning is to learn a hidden structure of given data (inputs). Note that such a *hidden structure* can be called as a *pattern* or a *concept*. This learning is useful for visualization of characteristics of data. Major algorithms in unsupervised learning include (1) *clustering* (which aims at dividing a set of inputs into groups but the groups are not known beforehand unlike in classification); and (2) *association rule learning* (which aims at discovering interesting relations among variables (inputs) in large databases).
- **Reinforcement learning**
  The goal of this learning is to learn a policy (which determines an appropriate action in a given situation) through a maximization of rewards provided from an environment. Reinforcement learning differs from the supervised learning because correct input/output pairs are never provided, *i.e.*, this learning should be completed to acquire an appropriate policy without a teacher who explicitly tells correct pairs..

Apriori algorithm

Among the above category of machine learning, the association rule learning categorized as the unsupervised learning has a great potential of correcting commonly accepted knowledge by providing the understandable knowledge extracted from the personal big data. Since the *apriori* algorithm is the major method among the association rule learning methods, this paper employs it to compare with our data mining method.

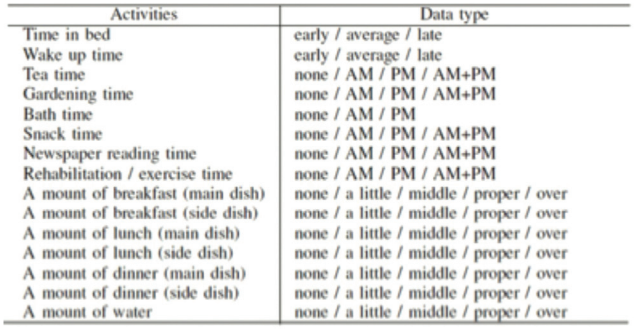
3. Care plan and daily activities

Care plan

In care houses, most of all aged persons want to have a comfortable and healthy life. For this issue, care houses try to provide the good appetite and proper rehabilitation to aged persons for their healthy bodies, and possibly their long life. To provide such an appropriate *lifestyle design* for aged persons, our previous research (Takadama 2014) developed the *concierge-based care support system* that supports aged persons by designing their own appropriate *care plans* (*i.e.*, rough schedules in a day) for a comfortable and healthy life. For example, a certain care plan starts from waking up, having meal, taking medicine, health check, exercise or rehabilitation, excretion, and sleep. What should be noted here is that (1) the current care plan is a common for all aged persons from the viewpoint of an efficient support for aged persons, which means that it may not be effective for a certain person; and (2) the current care plan is designed according to the experience of the care planner, which means that it has not yet perfectly optimized to each person.

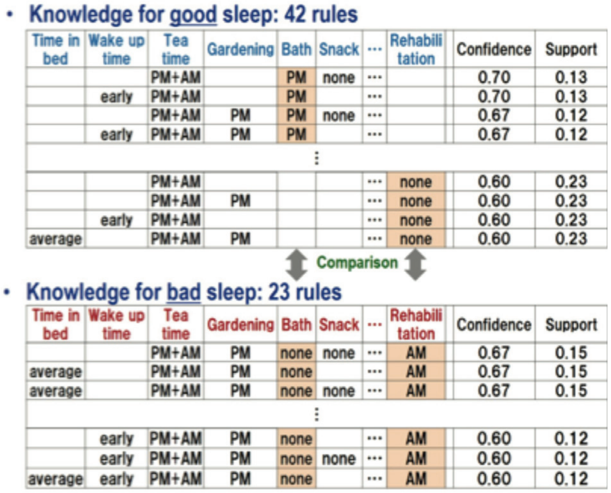
Dairy activities

Towards an appropriate care plan, we developed the novel data mining method (Takadama 2015) to extract essential daily activities (*e.g.*, rehabilitation, bathing) that contribute to deriving a deep/light sleep of aged persons. The detailed daily activities in care house are summarized as shown in Table 1. These activities are recoded (1) to extract essential ones that contribute to deriving a deep/light sleep of aged persons by the apriori algorithm and our method and (2) to compare the deepness of the sleep in the current care plan with one in the personalized care plan found by our method (hereafter, we call it the proposed care plan). Note that the daily activities that derive a light sleep are also important to be specified because the possibility of having a deep sleep increases by removing the activities that derive a light sleep.



Activities	Data type
Time in bed	early / average / late
Wake up time	early / average / late
Tea time	none / AM / PM / AM+PM
Gardening time	none / AM / PM / AM+PM
Bath time	none / AM / PM
Snack time	none / AM / PM / AM+PM
Newspaper reading time	none / AM / PM / AM+PM
Rehabilitation / exercise time	none / AM / PM / AM+PM
A mount of breakfast (main dish)	none / a little / middle / proper / over
A mount of breakfast (side dish)	none / a little / middle / proper / over
A mount of lunch (main dish)	none / a little / middle / proper / over
A mount of lunch (side dish)	none / a little / middle / proper / over
A mount of dinner (main dish)	none / a little / middle / proper / over
A mount of dinner (side dish)	none / a little / middle / proper / over
A mount of water	none / a little / middle / proper / over

Table 1 Daily activities in care house



- Knowledge for good sleep: 42 rules

Time in bed	Wake up time	Tea time	Gardening	Bath	Snack	···	Rehabilitation	Confidence	Support
		PM+AM		PM	none	···		0.70	0.13
	early	PM+AM		PM		···		0.70	0.13
		PM+AM	PM	PM	none	···		0.67	0.12
	early	PM+AM	PM	PM		···		0.67	0.12
				⋮					
		PM+AM				···	none	0.60	0.23
		PM+AM	PM			···	none	0.60	0.23
	early	PM+AM				···	none	0.60	0.23
average		PM+AM	PM			···	none	0.60	0.23

Comparison

- Knowledge for bad sleep: 23 rules

Time in bed	Wake up time	Tea time	Gardening	Bath	Snack	···	Rehabilitation	Confidence	Support
		PM+AM	PM	none	none	···	AM	0.67	0.15
average		PM+AM	PM	none		···	AM	0.67	0.15
average		PM+AM	PM	none	none	···	AM	0.67	0.15
				⋮					
	early	PM+AM	PM	none		···	AM	0.60	0.12
	early	PM+AM	PM	none	none	···	AM	0.60	0.12
average	early	PM+AM	PM	none		···	AM	0.60	0.12

Figure 1. Knowledge for a deep and light sleep (apriori algorithm)

4. Human subject experiments

Experimental design

To investigate the effectiveness of our data mining method by comparing with the apriori algorithm, our previous research (Takadama 2015) conducted the human subject experiments of the following three aged women in the actual care house: 82 aged diabetes person, 89 aged dementia and emotional illness person, and 107 aged healthy person. For an evaluation of the care plan, their sleep stage is investigated from the viewpoint of the deep/light sleep. Such an evaluation is based on a *personal big data* (*e.g.*, the heartbeat and body movement data) stored everyday by air-mattress sensor.

Experimental result s

- **Extracted knowledge (by apriori algorithm)**

  Fig. 1 shows some of the knowledge extracted by the apriori algorithm for a deep and light sleep of the 82 aged diabetes woman. Since the apriori algorithm generates a lot of knowledge through a calculation of all combinations of variables in data, the figure shows the only knowledge selected from 42 rules for a good sleep and 23 rules for a bad sleep, both of which are firstly selected with 60% confidence or more. As shown in this figure, the knowledge is based on many variables (*i.e.*, activities in this experiment). For example, the first knowledge for a good sleep suggests that a tea time in both AM and PM without snack and bathing contribute to have a deep sleep with 70% probability under 13% appearance in the dataset. Such a knowledge indicates some features for a deep sleep, but it is difficult for us to understand what kinds of activities are needed to have a deep sleep. If we carefully check the difference between knowledge for a good and bad sleep, we may notice that bathing or none of rehabilitation derives a good sleep while rehabilitation without taking a bath derives a bad sleep. This difference is significant but the apriori algorithm does not have a mechanism of providing such a difference.

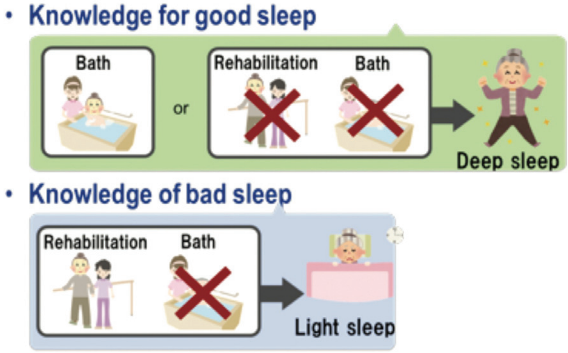


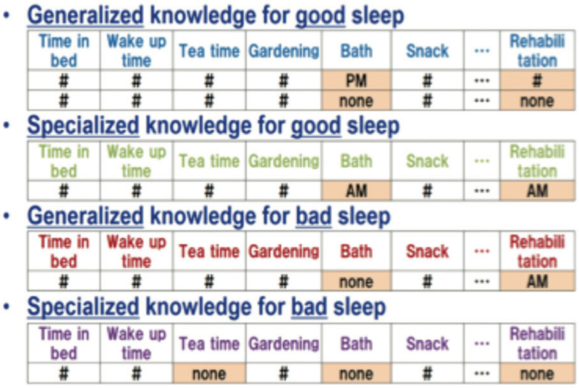
Figure 2 Knowledge for a deep and light sleep (out method) (Referring from (Takadama 2014) and (Takadama 2016))

- **Extracted knowledge (by our method)**

Fig. 2 shows the typical knowledge extracted by out method for a deep and light sleep of the same aged woman in the case of the apriori algorithm. This figure shows that the aged person has a good (deep) sleep when taking a bath or none of rehabilitation and bath, while the same person has a bad (light) sleep when undergoing rehabilitation without taking a bath. Note that the same relationship can be found in Fig 1. To understand this relationship, we interviewed the person and find that she always takes care of her body clean and she is willing to take a bath especially in the case of rehabilitation. From this interview, she can keep her body clean when taking a bath or not undergoing rehabilitation, which promotes her to have a comfortable sleep. In contrast, she cannot keep her body clean in the case of rehabilitation, which promotes her to have an uncomfortable sleep. By utilizing such knowledge, our proposed method can design the care plan which includes the daily activities that derive a deep sleep and excludes the daily activities that derive a light sleep. This kind of personalized care plan is not the common care plan among aged persons.

- **Detailed extracted knowledge (by our method)**

To investigate the above extracted knowledge in detail, Fig. 3 shows both of the generalized and specialized knowledge for a good (deep) and bad sleep. Concretely, our method found two generalized and one specialized knowledge for a good sleep, while one generalized and one specialized knowledge for a bad sleep. For example, the first knowledge of the generalized knowledge for a good sleep suggests that it is important to take a bath in P.M but it does not matter to need rehabilitation for a deep sleep (which represents with the # mark).



• Generalized knowledge for good sleep

Time in bed	Wake up time	Tea time	Gardening	Bath	Snack	...	Rehabili tation
#	#	#	#	PM	#	...	#
#	#	#	#	none	#	...	none

• Specialized knowledge for good sleep

Time in bed	Wake up time	Tea time	Gardening	Bath	Snack	...	Rehabili tation
#	#	#	#	AM	#	...	AM

• Generalized knowledge for bad sleep

Time in bed	Wake up time	Tea time	Gardening	Bath	Snack	...	Rehabili tation
#	#	#	#	none	#	...	AM

• Specialized knowledge for bad sleep

Time in bed	Wake up time	Tea time	Gardening	Bath	Snack	...	Rehabili tation
#	#	none	#	none	#	...	none

Figure 3. Generalized/specialized knowledge for a deep and light sleep (out method)

From these extracted knowledge, the following implications can be found:

(1) <u>Good vs. bad sleep knowledge</u>

A comparison of these four kinds of the extracted knowledge suggests that the essential daily activities in care plans are bathing and rehabilitation which are the same as shown in Fig. 2. Focusing on the generalized knowledge, in detail, the generalized knowledge for a *good* sleep suggests that the aged persons have a good sleep when taking a bath in PM or when neither taking a bath nor undergoing rehabilitation, while the generalized knowledge for a *bad* sleep suggests that the aged persons have a bad sleep when undergoing rehabilitation in A.M. without taking a bath.

(2) <u>Generalized vs. Specialized knowledge</u>

When comparing the generalized and specialized knowledge for a good sleep, the essential difference is a time of taking a bath, *i.e.*, the generalized knowledge suggests P.M. for a bath, while the specialized knowledge suggests A.M. for a bath. This is because a time of taking a bath of this aged person is generally assigned in P.M. in the current care plan, but its time changes to A.M. in the only case of her birthday party because the birthday party is scheduled to start in P.M. What should be noted here, however, is that such a small number of data is generally deleted as a noise by most of the data mining methods, but our method can keep such a small number of data which is indispensable to extract the specialized knowledge in addition to a large number of data which is used to extract the generalized knowledge.

(3) Specialized knowledge for a bad sleep
The specialized knowledge for a bad sleep suggests neither a bathing nor rehabilitation, but this knowledge matches the third generalized knowledge for a good sleep. What is the difference between these two knowledge is whether the aged person has or does not have a tea time. According to the interview to the aged person, she is diabetes person and is told from her doctor to take an appropriate amount of water. This gives an influence to her, that is, she can have a good sleep when she takes an appropriate amount of water, while she had a bad sleep when she cannot take an appropriate amount of water even in the condition of having a good sleep. This is because she is worry about it when she cannot have an appropriate amount of water, which derives a bad sleep. This result indicates that our method can extract the specialized knowledge acquired by only a few days when she cannot have an appropriate amount of water by missing a tea time due to some reasons.

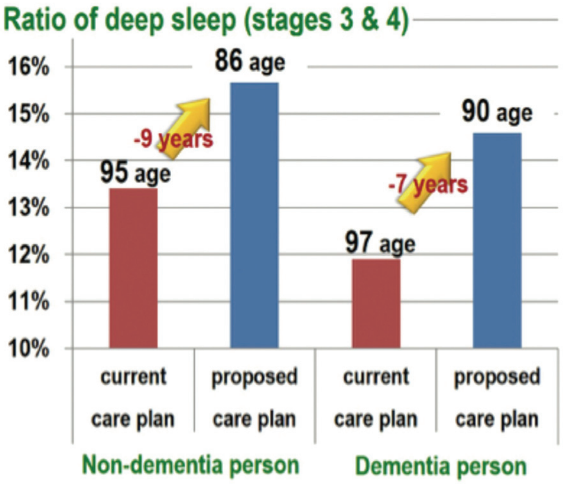


Figure 4. Deepness of sleep and care plans

(Referring from (Takadama 2014))

- **Effectiveness of extracted knowledge**

Finally, we compare the deepness of sleep of aged persons in the current care plan with that in the proposed care plan which includes the daily activities that derive a deep sleep and excludes the daily activities that derive a light sleep found by our method. Fig. 4 shows the deepness of the sleep, where the vertical and horizontal axes indicate the ratio of the sleep stages 3 & 4 and care plans, respectively. Note that the ratio of the sleep stages 3 & 4 increases in the case of a deep sleep while the ratio decreases in the case a light sleep. In this figure, the red and blue bars indicate the current and proposed care plans, respectively. In detail, the two bars from the left side indicate the results of the healthy aged persons (*i.e.* non-dementia aged person), while the two bars from the right side indicate the results of the dementia persons who are hard to have a deep sleep in comparison with non-dementia persons.

This figure shows that the ratio of the sleep stages 3 & 4 in the proposed care plan is higher than that in the current plan in both healthy (non-dementia) and dementia aged persons, which means that the proposed care plan can provide a deep sleep in comparison with the current care plan. This is very important in care houses because such a deep sleep contributes to decreasing a frequency of wandering in midnight. From the viewpoint of the *sleep age* estimated from the ratio of sleep stages 3 & 4 averaged from a lot of aged persons (The Japanese Society of Sleep Research, 2010), the proposed care plans can provide nine years younger sleep in the healthy aged persons and seven years younger sleep in dementia persons in this experiment. What should be noted here is that the proposed care plans have a great potential of providing younger sleep even in dementia persons, although they are generally difficult to have a deep sleep.

5. Conclusion

This paper focused on care support knowledge (especially focuses on the sleep related knowledge) and tackled its *cognitive bias* and *humanity* aspects from machine learning perspective through discussion of whether machine learning can correct commonly accepted knowledge and provide understandable knowledge in care support domain. For this purpose, this paper started by introducing our data mining method (based on association rule learning) that can provide only necessary number of understandable knowledge without probabilities even if its accuracy slightly becomes worse, and shows its effectiveness in care plans support systems for aged persons as one of healthcare systems. The experimental result indicates that (1) our method can extract a few simple knowledge as understandable knowledge that clarifies what kinds of activities in care house contribute to having a deep sleep, but (2) the *apriori* algorithm is hard to provide such knowledge because it needs calculate all combinations of activities executed by aged persons.

What should be noted here is that the above potentials have only been shown from only one aged person. This suggests that further careful qualifications and justifications, such as an investigation of other aged persons, are needed to generalize the proposed rough guideline. Such important directions must be pursued in the near future in addition to the following future research: (1) an analysis of other days or other period; (2) a long-term analysis of effect of the personalized care plan; and (3) a deep discussion of *cognitive bias* and *humanity* for well-being computing.

Acknowledgments

The research reported here was supported in part by a Grant-in-Aid for Scientific Research (A, No. 15H01720) from the Ministry of Education, Culture, Sports, Science and Technology (MEXT), Japan. The author thanks Yusuke Tajima for preparing figures in this paper.

References

Agrawal, R, Imielinski, T. and Swami A. 1993. "Mining association rules between sets of items in large databases", The 1993 ACM SIGMOD International Conference on Management of Data, pp.207-216.

Agrawal, R.and Srikant, R. 1994. "Fast algorithms for mining association rules", The 20th Int. Conf. Very Large Data Bases, VLDB, pp.487-499.

Gilovich, T., Griffin, D., and Kahneman, D. (Eds.) 2002. *Heuristics and Biases: The Psychology of Intuitive Judgment*. New York: Cambridge University Press.

Haselton, M. G., Nettle, D., and Andrews, P. W. 2005. *The evolution of cognitive bias*. In D. M. Buss (Ed.), The Handbook of Evolutionary Psychology: Hoboken, NJ, US: John Wiley & Sons Inc. pp. 724–746.

Kahneman D., Slovic P., and Tversky, A.（Eds.）1982, *Judgment Under Uncertainty: Heuristics and Biases*. New York: Cambridge University Press.

Kahneman, D., Krueger, A. Schkade, D., Schwarz, N., and Stone, A. 2006. "Would you be happier if you were richer? A focusing illusion," *Science*, Vol. 312, No. 5782, pp. 1908–1910.

Mitchell, T. 1997. *Machine Learning*, McGraw Hill.

LeCun, Y., Bengio, Y., and Hinton, G. E. 2015. "Deep Learning", *Nature*, Vol. 521, pp. 436-444.

Hinton, G. E., Osindero, S., and Teh, Y. 2006a. "A fast learning algorithm for deep belief nets," *Neural Computation*, Vol. 18, pp. 1527-1554, 2006.

Hinton, G. E. and Salakhutdinov, R. R. 2006b "Reducing the dimensionality of data with neural networks," *Science*, Vol. 313. No. 5786, pp. 504-507.

Rumelhart, D. E., Hinton, G. E., and Williams R. J. 2986. "Learning internal representations by error propagation," *Parallel Distributed Processing*, MIT Press, Vol. 1, pp. 318-362.

Takadama, K. 2014. ``Concierge-based Care Support System for Designing Your Own Lifestyle," *The AAAI 2014 Spring Symposia, Big Data Becomes Personal: Knowledge into Meaning*, pp. 69-74.

Takadama, K. and Nakata, M. 2015. ``Extracting Both Generalized and Specialized Knowledge by XCS using Attribute Tracking and Feedback," *2015 IEEE Congress on Evolutionary Computation (CEC2015)*, pp. 3034-3041.

Takadama, K. 2016. ``Well-being Computing Towards Health and Happiness Improvement: From Sleep Perspective," *The AAAI 2016 Spring Symposia, Well-Being Computing: AI Meets Health and Happiness Science*, pp. 417-422.

The Japanese Society of Sleep Research. 2010. Handbook of sleep science and sleep medicine, Asakura Publishing Co., Ltd., p.27-29.

Study of Analytical Methods on the Relationship between Sleep Quality and Stress with a Focus on Human Circadian Rhythm

Ryo Takano,[†] Satoshi Hasegawa,[†] Yuta Umenai,[†] Takato Tatsumi,[†] Keiki Takadama,[†] Toru Shimuta,[‡] Toru Yabe,[‡] Hideo Matsumoto[‡]

[†]Graduate School of Informatics, The University of Electro-Communications, Tokyo, Japan
[‡]Murata Manufacturing Company, Ltd.
takano@cas.hc.uec.ac.jp, umenai@cas.lab.uec.ac.jp, tatsumi@cas.lab.uec.ac.jp, keiki@inf.uec.ac.jp
t-yabe@murata.com, shimuta@murata.com, hmatsumoto@murata.com

Abstract

The purpose of this study is to find novel knowledge to clarify the relationship between the sleep quality and the degree of the mental stress. For this purpose, we focus on not only these two indices (the quality of sleep and the degree of the mental stress), but also the human circadian rhythm as the new index for analysis. Through three types of data measured during the night-time sleep and during the day, we tried to inspect the usefulness of the human circadian rhythm for the index of the analysis. In this paper, data of these three indices were measured by the single subject experiment of about two weeks and analyzed comprehensively. In the analysis, we categorize good / middle / bad for each index every few days, and investigating the relationship between the three indices by summarizing the transition of the categories of the three indices. As a result, by comparing three types of data of ten-odd days in parallel, we obtained the following findings: (1) These three indices have been moving with a similar trend in units of days; (2) those trends coincide details from the simple diary written by the subject. As a result, by comparing three types of data of ten-odd days in parallel, these data were related to each other.

Introduction

The goal of this study is to clarify the relationship between the quality of sleep and the degree of stress. It goes without saying that the main function of sleep is recovery of fatigue of the body. And similarly, sleep has the effect of restoring mental stress [Grant, 2010]. For these reasons, a good quality of sleep is the important factors for maintaining physical and mental health. However, since sometimes people with high stress often fail to sleep well, the relationship between the two indices is complicated and still unknown [Grant, 2010]. For this reason, how to get a good quality of sleep aimed is one of the important tasks in study for caring health and mental health. However, attempts to clarify the relationship between the quality of sleep and the degree of mental stress have not yet been fully achieved.

In this study, we focus on human circadian rhythm as the new viewpoint for analysis between the quality of sleep and the degree of mental stress. And we try to obtain new findings on this relationship between the quality of sleep and the degree of stress by analysis with the human circadian rhythm. Circadian rhythm is known to be an important factor in determining the quality of sleep [Robert, 2007]. Moreover, when this circadian rhythm cause to slip out from actual life rhythm consisting of a cycle of sleep and action, human fall into a state such as jet lag, and high stress is likely to occur. Accordingly, this study expects that using this circadian rhythm for analysis is to be able to connect two indices with different observation periods "quality of sleep measured in the night" and "stress measured during the day". As described above, we verify that new relationships be able to obtain from three kinds of data "quality of sleep measured in the midnight", "stress measured during the day" and "circadian rhythm measuring day and night".

The remaining of this paper is organized as follows. First, second section shows the proposed method for analysis of this study. Third section shows the subject experiment and its result. Finally, the conclusion of this paper is given in the final section.

Analytical method

In this chapter, we explain three types of data for this study. These three types of data are "quality of sleep measured in the midnight", "stress measured during the day" and "circadian rhythm measuring day and night". Moreover, the comprehensive analysis method by these three types of data is also described in this chapter. The comprehensive analysis method reveals the changes in daily physical condition by summarizing these three type of data.

The content of the usage data

Details of three types of data used for the analysis is explained respectively. These three types of data are indices to measure "quality of sleep", "degree of stress" and "Goodness of Human circadian rhythm ", respectively.

Estimated sleep stage

First, as the index of "quality of sleep" we employed the estimated sleep stage. The sleep stage is represented by a numerical value meaning the depth of sleep. The estimated sleep stage is determined according to the characteristics of vital data obtained from an air mattress biosensor without connecting any devices to human body and can roughly estimate the sleep stage without medical experts [Harada 2016]. The estimated sleep stage is classified into six types as follows: “Wake”: awake as the lightest sleep; “Rem”: REM sleep which is deeper sleep than awake and it is generally occurred in dreaming time; NonRem1-4: Non-Rem sleep 1 - 4 as deep sleep (note that the depth of sleep becomes deeper from NonRem1 to NonRem4). In this study, we employed the proportion of the total time of Wake, Rem and NonRem1 in the six stages as the index of the quality of sleep. The smaller the ratio is shown that the longer the deep sleep is maintained, and the quality of sleep is good. In Figure 1, the proportion of the estimated sleep stage in a night is shown as the example.

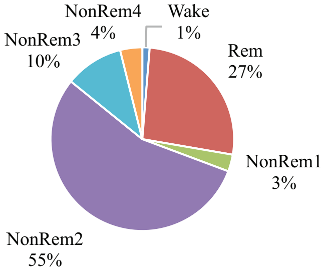


Figure.1 The proportion of the estimated sleep stage

Index of Stress degree

Second, as the index of "the degree of stress" we employed "LF / HF". "LF / HF" is the index showing the balance between the sympathetic nerve (LF) and the parasympathetic nerve (HF), which means that the higher the value show that a person gets the higher the stress. Figure 2 shows the transition of LF / HF measured every few hours for two days as the example. When the value of LF / HF is smaller than 2, the stress is small. On the other hand, when it is 2 or more, it indicates that the stress is high.

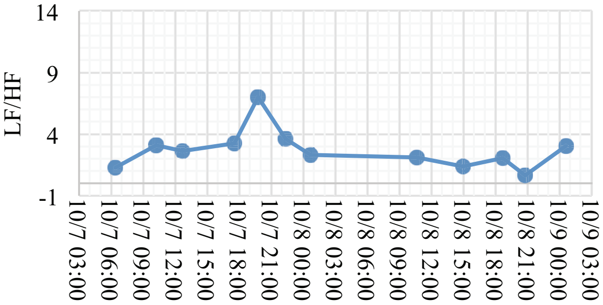
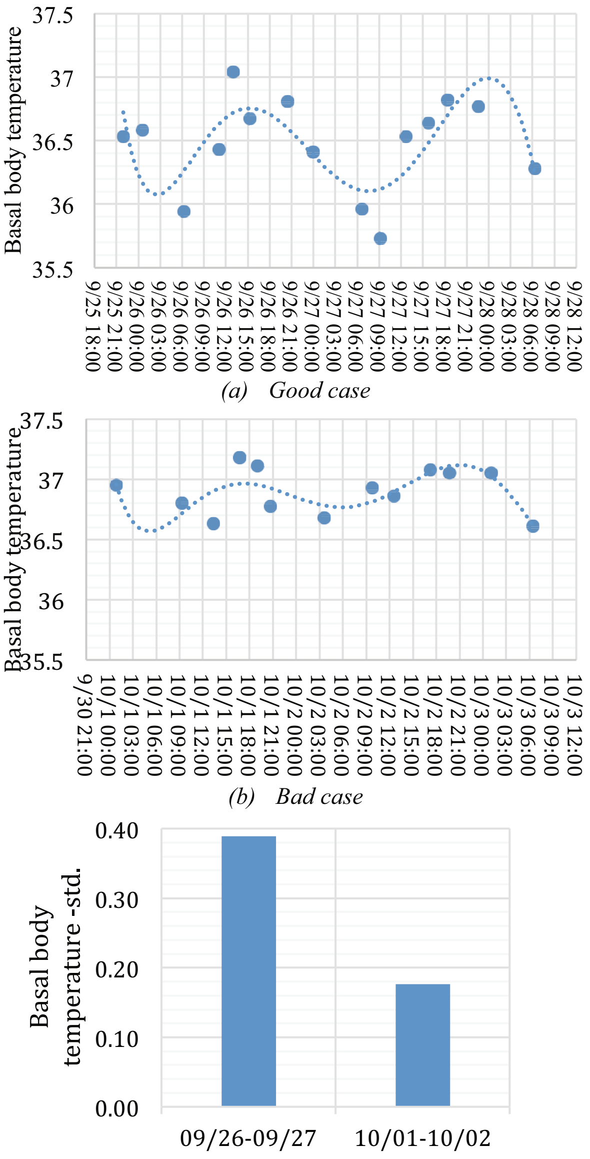


Figure.2 The transition LF / HF for two days

Human circadian rhythm

Thirdly, as the index showing "goodness of Human circadian rhythm" we employed the standard deviation of Basal body temperature for two days. It is known that basal body temperature goes up and down according to circadian rhythm. Based on this knowledge, by observing the standard deviation of basal body temperature during the measurement period, it can be observed whether the circadian rhythm holds a definite pattern. The definite pattern of circadian rhythm is an index related to good sleep and good activity during two days.

The concrete example is shown in Figure 3 which consist of 3 graphs (a) to (c). In Figure 3, actual basal body temperature during two days (dot) and its sixth order polynomial approximation (dash line) has been drawn in 2 of 3 graphs ((a-b)). In these graph, the dash line can express as a pattern of circadian rhythm during two days. Figure 3 (a) is good pattern of circadian rhythm, because the difference in height between the two sets "Valley at late night" and "Peak at afternoon" of dash line is large, so that the pattern can be clearly seen. On the other hand, figure 3 (b) is bad pattern, because the difference in height between "Valley and Peak" is small, so that the pattern can be unclearly seen. In order to evaluate the good or bad of the pattern judged from these graphs only with numerical data, we employ calculating the standard deviation of the basic body temperature. In figure 3 (c), the standard deviations of the basic body temperature of figure3 (a) and (b) are shown in. As can be seen from figure 3 (c), the standard deviation of 9/26 - 9/27 showing the good pattern in figure 3 (a) shows higher numerical value than that of 10/1 - 10/2 which showed the bad pattern in figure 3 (b). In this way, it is possible to discriminate the quality of circadian rhythm based on the magnitude of the standard deviation of the basal body temperature.



(a) Good case

(b) Bad case

(c) Standard deviation of every two days

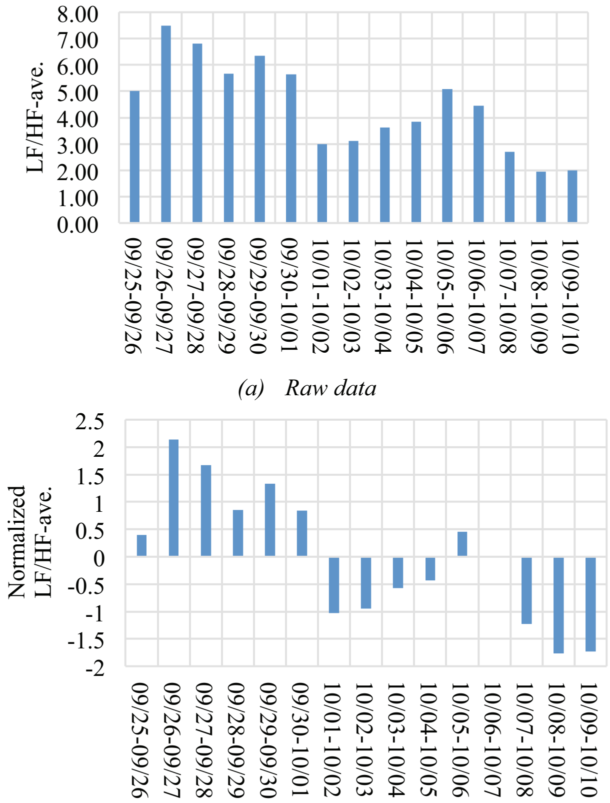
Figure.3 The circadian rhythm from basal body temperature

Comprehensive analysis

The analysis aims to clarify the causal relation between sleep quality and stress from the measurement result as 3 types of data "the sleep stage, LF / HF and basal body temperature" about 2 weeks' period. In this section, we describe the comprehensive analysis using 3 type of data. For this purpose, prepare the proportion of shallow sleep per night as the data of the estimated sleep stage. For the other two types of data, it is calculated the average and standard deviation for every two days by the values which measured every several hours. The reason for summarizing these two days is to observe numerical values before and after the data during the two days of sleep.

At first, for two data other than estimated sleep stage data, numerical standardization is carried out in order to improve the ease of visual analysis. Numerical standardization is calculated as that the values subtracted each numerical value by the own average and divided by the own standard deviation. As a result, the average value and the standard deviation of the data set is 0 and 1, respectively. And the closer the value in the data is 0, the closer it is to the mean value. On the other hand, large value in the data is as characteristic data. Specifically, the data is standardized as shown in figure 4. Figure 4 (a) shows average values of LF / HF before normalization for about 2 weeks, and figure 4 (b) shows the normalized average values of LF / HF for same period. Compared with figure 4 (a), it is possible to discriminate the level of stress simply by looking at figure 4 (b) at a glance.

Next, we describe how to analyze three types of data. In order to understand each cause and effect relation, three types of data are classified according to good or bad with units of several days in the same time axis, respectively. In this way, we consider the relationship between sleep quality, stress magnitude and circadian rhythm from this classification and the simple diary by subjects. The specific example is to introduce in the next chapter.



(a) Raw data

(b) Normalized data

Figure.4

Human subject experiment

In this section, as first step of whether it is possible to obtain novel knowledge about the relationship between the quality of sleep and the degree of stress with a focus on human circadian rhythm, actual measurement data is analyzed using the three indices described in the previous chapter.

Details of usage data

The usage data are values of the three indices of adult females measured during the 15 days from September 25, 2016 to October 10, 2016. The Estimated sleep stage is calculated by the data for one night. For measuring instruments, EMFIT's mat sensor was used to calculate sleep stage, stress measurement sensor (VM 302) of Fatigue Science Laboratory Inc. for measuring LF / HF, and commercially available basic thermometer was used for measurement of basal body temperature. And LF / HF and basal body temperature were measured every few hours in a day. After the end of the measurement, we calculated the average of the LH / HF and the standard deviation of the basal body temperature every 2 days. In addition, these value of LF / HF and the basal body temperature is rendered with a normalized value so that the average is 0 and the standard deviation is 1. In addition, we asked the subjects to record a simple diary. The summary of the contents of the diary is shown in Table 1.

Table.1 Short diary by the subject

9/25	No description
9/26	Business meeting Exercise at the gym
9/27	Not have lunch
9/28	Leave the office regularly
9/29	Business seminar Drinking party
9/30	Business trip Important negotiation
10/1	[Holiday] Wander around Kanazawa
10/2	[Holiday] Chat with a friend
10/3	Desk work in all day
10/4	Long meeting Exercise at the gym
10/5	Go home early
10/6	Exercise at the gym
10/7	Busy day
10/8	[Holiday] Relaxing at home
10/9	[Holiday] Shopping

Result of measurement

First, the measurement results obtained in the experiment are shown in follows: the estimated sleep stage is shown in figure 5; the normalized average of LF / HF is shown in figure 4 (b); and the normalized standard deviation of basic body temperature is shown in figure 6. For the graph of the proportion of the estimated sleep stages in figure 5, for the subsequent analysis, the values below the 25th percentile and above the 75th percentile are color coded into blue and red, respectively. This is to make it easy to distinguish between the good quality sleep, the middle quality of sleep and the bad quality of sleep.

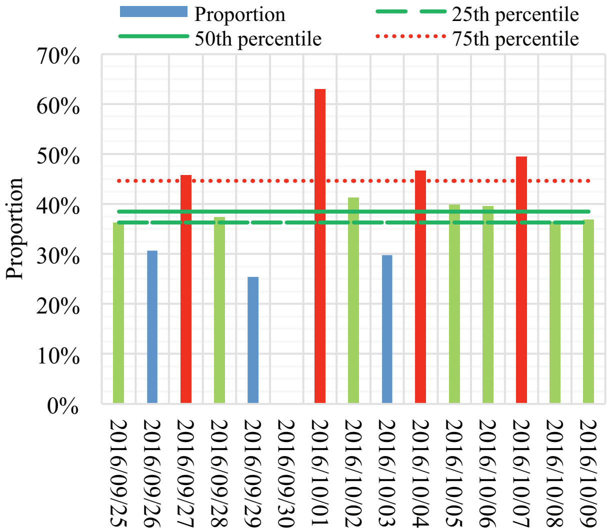


Figure.5

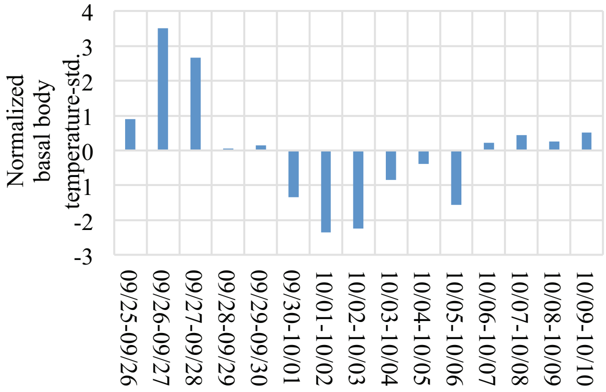


Figure.6

Analysis

In this section, as described in the previous chapter, we analyze the relationship by classifying each data by the good or bad. In this analysis, it is conducted for each data the case of categorizing into two "good or bad" and the case of categorizing into three "good, middle or bad". The reason for analyzing two cases, "the case of categorizing into two" and "the case of categorizing into three", is to verify whether there is a difference in analysis result depending on the number of categories.

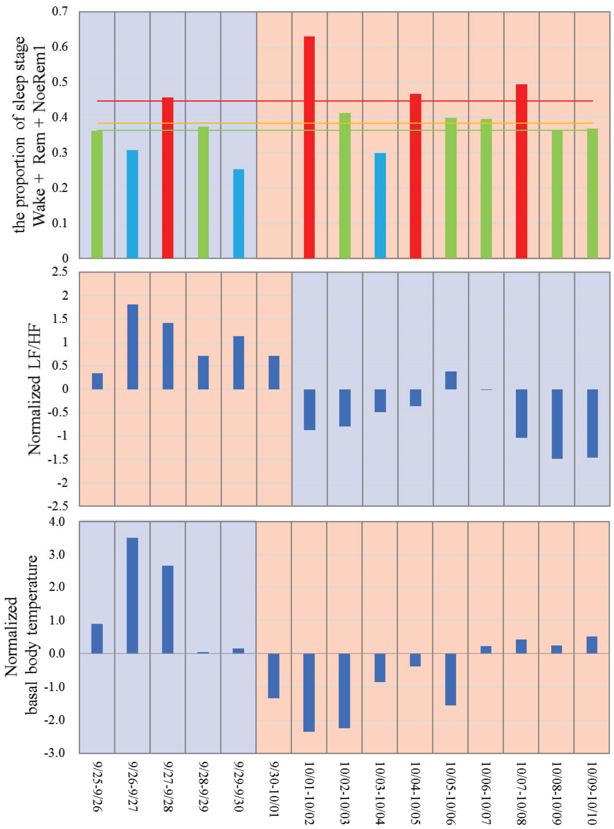


Figure.7 Result of three indices

Classified into two -Good or Bad-

Figure 6 shows the numerical values of each indices calculated and the graphs obtained coloring them according to the degree of goodness. Figure 6 is composed of three bar graphs, with the first row showing sleep stage, the second row showing LF / HF and the third row showing the standard deviation of basal body temperature. In addition, the color of each day bar in the upper graph classifies each sleep by blue, green and red as three types of "quality of sleep" (good or bad). The color of the background in Figure 6 indicates the goodness-badness of the indices, as viewed in a few days' span. First, when comparing the sleep stage of the upper row and the standard deviation of basal body temperature of the lower row, the transition of the color tone of the background coincides. Next, when comparing the LF/HF of the middle row and the standard deviation of basal body temperature of the lower stage, the color of the background shades of red and blue are reversed with one day's deviation. Focusing on these, the trends of all indices have changed significantly from September 30 to October 1. From short diary of the subject (Table 1) about September 30, it found that she is on business trip and carrying out important negotiations. And it also found that she did not measure sleep because she stayed at the hotel on this day.

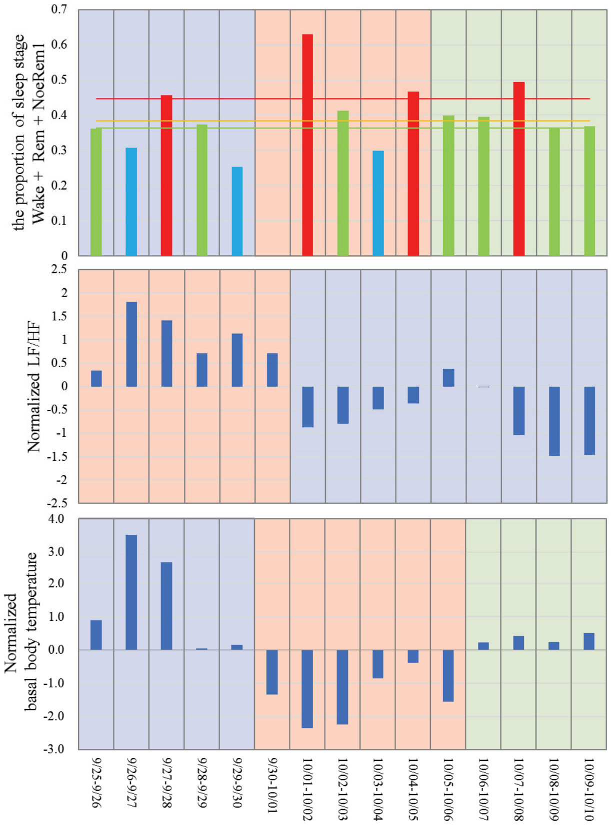


Figure.7 Result of three indices

Categorize into three - Good, Middle or Bad-

Figure 7 shows the numerical values of each indices calculated and the graphs obtained coloring them according to the degree of goodness. Figure 7 is composed of three bar graphs, with the first row showing sleep stage, the second row showing LF / HF and the third row showing the standard deviation of basal body temperature. In addition, the color of each day bar in the upper graph classifies each sleep by blue, green and red as three types of "the quality of sleep" (good, middle or bad). The color of the background in Figure 7 indicates the goodness-badness of the indices, as viewed in a few days' span. First, when comparing the sleep stage of the upper row and the standard deviation of basal body temperature of the lower row, the transition of the color tone of the background coincides with one day's deviation. Next, when comparing the LF/HF of the middle row and the standard deviation of basal body temperature of the lower stage, the color of the background shades of red and blue are reversed. These results are consistent with the results of the categorized into three shown in figure 6. According to the sleep stage of the upper row and the standard deviation of basal body temperature of the lower row in the graph, both of the middles areas indicated by the green background almost much with each other.

Summary

From these results of analysis in 2 cases of categorization, she felt stress during several days for important negotiations, but stress has been relieved after the negotiations. Furthermore, to recover from stress in this several days, it is considered that the circadian rhythm became a large pattern and led to deep sleep.

Conclusion

The final goal of this study is to clarify the relationship between sleep and stress. For this purpose, we focus on the human circadian rhythm as the new index. Using this circadian rhythm for analysis is expected to be able to connect two data with different observation periods "quality of sleep measured in the night" and "stress measured during the day". Through three types of data measured during the night-time sleep and during the day, we tried to verify the usefulness of this focus on human circadian rhythm.

In order to verify the effectiveness of this point of view, we conducted subject experiments for 15 days. In this experiment, the subject's sleeping data, basal body temperature and HF / HF were measured during the period. From these, three indices of sleep quality (sleep stage), degree of stress, goodness of circadian rhythm pattern were calculated. By comprehensively analyzing these three indices, we obtained the following findings: (1) These three indices have been moving with a similar trend in units of days; (2) those trends coincide details from the simple diary written by the subject. As a result, by comparing three types of data of ten-odd days in parallel, these data were related to each other. As a result, by comparing three types of data of ten-odd days in parallel, these data were related to each other. As a future task, we verify the versatility of this analysis method by more subject experiments.

References

Harada, T.; Uwano, F.; Komine, T.; Tajima, Y.; Kawashima, T; Morishima, M.; and Takadama, K, 2016. Real-time Sleep Stage Estimation from Biological Data with Trigonometric Function Regression Model. In *the AAAI 2016 Spring Symposia, Well-Being Computing: AI Meets Health and Happiness Science, AAAI The Association for the Advancement of Artificial Intelligence)*, pp. 348-353.

Grant, B., 2010. Sleep: an important factor in stress-health models. *Stress and Health*. Volume 26, Issue 3: 204–214

Robert, L. S.; Dennis, A.; R. Robert, A.; Mary, A. C.; Kenneth, P. W., Jr; Michael V. V. and Irina V., Z. 2007. Circadian Rhythm Sleep Disorders: Part I, Basic Principles, Shift Work and Jet Lag Disorders. *Sleep*, Volume 30, Issue 11: 1460-1483

W., Jr; Michael V. V. and Irina V., Z. 2007. Circadian Rhythm Sleep Disorders: Part II, Advanced Sleep Phase Disorder, Delayed Sleep Phase Disorder, Free-Running Disorder, and Irregular Sleep-Wake Rhythm. *Sleep*, Volume 30, Issue 11: 1484-1501

Improving Sleep Stage Estimation Accuracy by Circadian Rhythm Extracted from a Low Frequency Component of Heart Rate

Akari Tobaru, Fumito Uwano, Takuya Iwase, Kazuma Matsumoto,
Ryo Takano, Yusuke Tajima, Yuta Umenai and Keiki Takadama
The University of Electro-Communications,
1-5-1, Chofugaoka, Chofu, Tokyo, Japan
{tobaru_akari,uwano,tanu_iwa,kazuma,takano,y_tajima,umenai}@cas.lab.uec.ac.jp, keiki@inf.uec.ac.jp

Abstract

This paper described that proposing a novel method to estimate the sleep stage by biological data obtained with a non-contact sensor devices and that investigating its effectiveness. Proposed method focused on circadian rhythm to consider of a day biological rhythm in overall sleeping in addition to employ the Haradas method. To verify the effectiveness of the proposed method, we derived the subject experiment that compared with the evaluation accuracy by the previous method with adjustment of circadian rhythm. As the experimental results, the following implications have been revealed: (1) the accuracy of the sleep stage estimation in 5 days out of that in 9 days were improved by proposed method in comparison with Haradas method; (2) the parameter β (which determines the discount rate of curve of circadian rhythm) should be set around 60%, meaning that a raw circadian rhythm (i.e., no discounted rhythm) strongly affected the sleep stage while the highly discounted circadian rhythm (e.g. 30% discounted rhythm) does not contribute to accurately estimating the sleep stage.

Introduction

Recently, sleep disorders affect our health condition, which causes a bad influence to human activities in our societies. According to the survey of the Ministry of Health, Labor and Welfare of Japan (Ministry of Health, Labor and Welfare 2015), one out of five adults in Japan suffer from insomnia, and the number of those who answered that they cannot take sufficient sleep has been increasing year by year.

To examine sleep disorder including insomnia, the Rechtschaffen & Kales method (hereinafter referred as the R & K method) is widely employed as the international standards method in the current medical field. The R& K method can estimate the sleep stage with a high accuracy based on the doctor's experience and knowledge, however it gives mental or physical burden to patients because many electrodes should be attached to the patient's face and head in order to gain electroencephalogram (EEG), electrooculogram (EOG), and electromyogram (EMG) data. To tackle this problem, the several non-contact sleep stages estimation methods have been proposed in recent years. However, the essential problem of such conventional non-contact methods is that the accuracy of the sleep stage estimation is not high enough in comparison with R & K method.

To overcome this problem, this paper proposes a novel non-invasive sleep stage estimation method based on circadian rhythm which is a human biological rhythm of one day. We focus on circadian rhythm because none of the conventional methods do not take it into account even though it indirectly affects the sleep stage. For this reason, this paper starts to improve Haradas method (Real-time Sleep Stage Estimation, hereafter referred to RSSE) (Harada and Takadama 2017) by introducing circadian rhythm to improve the accuracy of the sleep stage estimation. We employ Haradas method as the baseline of the sleep stage estimation because its accuracy is higher than other conventional methods and it can estimate the sleep stage in real time which is more difficult than in all time during sleeping. While circadian rhythm is basically measured by the body temperature changes, proposed method estimates circadian rhythm from a low frequency component of heart rate because of a correlation between body temperature and heart rate. As the main difference between Haradas method and proposed method, the former method estimates the sleep stage on the bases of an average of medium frequency component of heart rate, while the latter method estimates the sleep stage on the bases of a low frequency component of heart rate as circadian rhythm.

This paper is organized as follows. The next section explains some conceptions with regard to the measurement of sleep. In section of Related works, we introduce some related works, especially focus on non-contact sleep stage estimation methods. Section of RSSE explains the previous method which we employ as a base line and section of Sleep Stage Estimation Method Based on Circadian Rhythm (RSSE-CR) describes how to take circadian rhythm into account and processing for the weighting wave of circadian rhythm. Section of Experiment explains the subject experiment in order to verify the effectiveness of the suggested idea and the result is shown in section of Result, then we discuss what is effective to estimate sleep stage in section of Discussion.Finally, we conclude this paper and indicate future works in section of Conclusion.

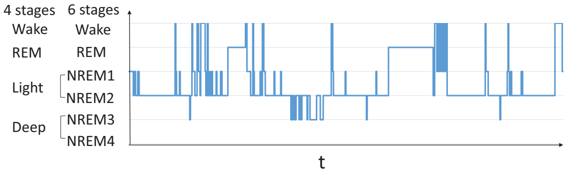


Figure 1: A graphical image of the sleep stage estimation

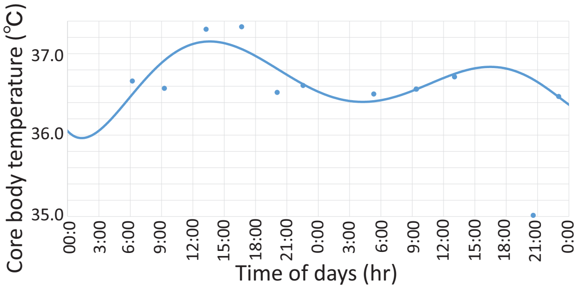


Figure 2: An illustration of an example for circadian rhythm

Sleep and its rhythm

Sleep stage

According to R&K method, sleep stage is divided into six parts called Wake, REM, NREM1, NREM2, NREM3, NREM4 toward deep sleep which is available to digitize sleeping objectively as Figure1 shows. Beginning to sleep, human tends to get deepest sleep immediately in general, then repeating the sleep cycle called ultradian rhythm. In an entire view, it becomes light sleep as the time to wake up is coming. Recently, the evaluation with four sleep stage became more general, which replaces that NREM1 and NREM2 for Light-sleep, and that NREM3 and NREM4 for Deep-sleep.

Circadian rhythm

Circadian rhythm is normally a twenty-four hours cycle, which effects physiological processes such as sleeping and waking. Circadian rhythm is represented by core body temperature as Figure2 shows. With changes of inter- nal body temperature, physical activity of animals including human is controlled by circadian rhythm. For example, an- imals get to feel sleep at three to four o clock when cir- cadian rhythm is to be the lowest point, also at the evening when it is to be the highest point, animals get to be active. Conversely, disturbing of circadian rhythm affects physical processes, some of which are jet lag, insomnia.

Related works

The several non-invasive sleep stages estimation methods have proposed in recent years. The examples includes (i) the two-stage sleep estimation method by an infrared motion sensor or microwave radar which can classify deep or light in the sleep stage (Kamibayashi and Hagiwara 2012), (ii) the REM/non-REM classification method based on breathing extracted from microwaves sensor(Sasaki et al. 2015), (iii) the machine learning estimation method based on the biological signals (Komine et al. 2016), and (iv) the sleep stage estimation based on heartbeat and body movement measured by the air mattress sensor (Watanabe and Watanabe 2002)(iv-1),(Harada and Takadama 2016)(iv-2). Specifically, the previous method(RSSE) researched by Harada and Takadama employed the EMFit sensor made by the VTT Technical Research Center in Finland as a non-contact biosensor in order to obtain biological data such as heart rate and body movement every second. As Figure3 shows, the EMFit is used by being laid under a bedmattress. Table1 is what classified these previous methods to estimate sleep stage by category of both used biological data and measurement time, and we focus on (iv-2) at the lower left on the table1.

Figure 3: The EMFit sensor being laid under a bed mattress

Table 1: Classification of previous methods

	HR+BM	BM only
Non-realtime	(iii),(iv-1)	(i)
Realtime	(iv-2)	(ii)

Real-time Sleep Stage Estimation(RSSE)

RSSE is constructed by the process as following: (1) to obtain biological data from the mattress sensor (shown by Figure3; (2) to calculate the medium frequency component of the heart rate based on the regression of the trigonometric function (referred to in (step a)); (3) to standardize the calculated medium frequency; (4) to discretize the medium frequency (referred to in (step b)) and (5) to compensate the evaluation of Wake and REM (referred to in (step c)). These processes are dealt with in sequence depended on numbering.

(step a)Trigonometric function regression model RSSE acquired medium frequency component of the heart rate by the composed wave of many trigonometric functions approximate an intermediate frequency as follows.

$$h(t,\phi) = c + \sum_{n=1}^{N} a_n \cos(\frac{2\pi t}{L/n}) + b_n \sin(\frac{2\pi t}{L/n}) \tag{1}$$

In this formula, N is determined as the parameter and L denotes the maximum data for approximation, which are set as following $N = 13$,$L = 2^{14}$ in this paper. Also ϕ is a set of parameters $\{a_1, b_1, ..., a_n, b_n\}$ and both a_n and b_n indicate the coefficients of a cosine wave and sine wave amplitude for each one, which are the trigonometric function with setting a cycle to L/n. These the trigonometric functions are calculated to make an error of $J(\phi)$ smallest in comparison between the raw heart rate and the calculated approximation of heart rate. In formula(2), λ is set as $\lambda = 1$, c denotes a constant and t indicates time.

$$J(\phi) = \frac{1}{T}\sum_{t=1}^{T}(HR(t) - h(t,\phi))^2 + \frac{\lambda}{N}\sum_{n=1}^{N}(a_n^2 + b_n^2) \tag{2}$$

(step b)Discretization After calculating the approximated medium frequency component of heart rate $h(t, \phi)$, sleep stage is estimated by discretizing depended on the following formula:

$$s(t) = \begin{cases} 5 & \lceil \frac{(h(t,\phi)-ave.)}{stdev.} + 2 \rceil > 5, \\ 0 & \lceil \frac{(h(t,\phi)-ave.)}{stdev.} + 2 \rceil < 0, \\ \lceil \frac{(h(t,\phi)-ave.)}{stdev.} + 2 \rceil & \text{otherwise.} \end{cases} \tag{3}$$

$$ave. = \frac{1}{\max(T,L)} \sum_{t=1} \max(T,L) h(t,\phi) \tag{4}$$

$$stdev. = \sqrt{\frac{1}{\max(T,L)-1} \sum_{t=1} \max(T,L)(ave. - h(t,\phi))^2} \tag{5}$$

$s(t)$ denotes the sleep stage at time t, $\lceil x \rceil$ indicates the ceiling function that returns the minimum integer value equal to or greater than x, and from 5 to 0 correspond to Wake, REM, NREM1,NREM2,NREM3,NREM4 respectively, This discretization formula is based on the previous research (Takadama et al. 2010).

(step c)Wake/REM Classification For Wake classification, RSSE focuses on a huge body movement in sleeping. To consider a variability in the body movement, it calculates both the standard deviation BM_{std} in every minute and the average of the body movement BM_{ave} from start of sleeping, then classified the sleep stage into Wake over the latest one minute by the function is defined as follows :

$$\frac{BM_{std}}{BM_{ave}} > 1.0 \tag{6}$$

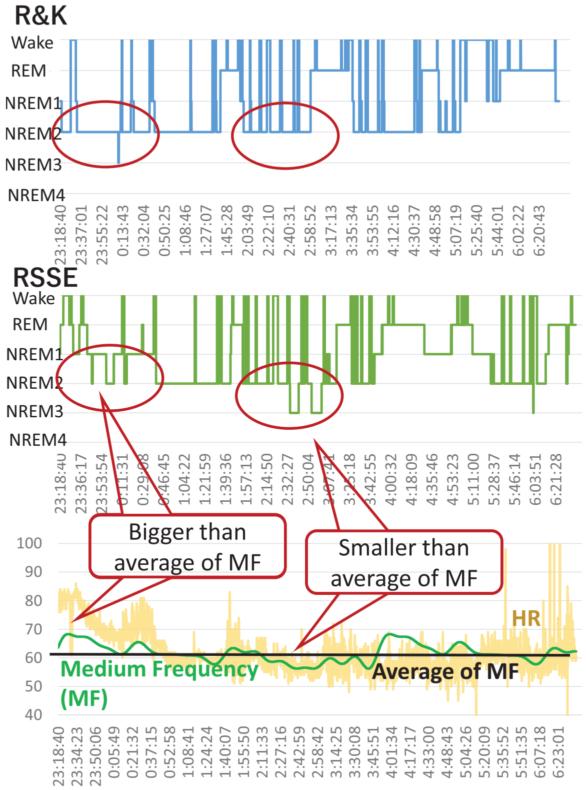


Figure 4: The process to evaluation the sleep stage in RSSE

For REM classification, the rapid heart rate variability is employed. The start point of REM is set if the growth rate in median value of heart rate during the last five minutes is greater than 4 % compared with the growth rate from the last ten minutes to the before five minutes, which is represented by the formula (7). In addition, the condition of REM is canceled if a huge body movement occurred within near ten minutes or avoiding the misclassification.

$$\frac{(HR_{med}^{recent} - HR_{med}^{prev})}{HR_{med}^{prev}} > 0.04 \tag{7}$$

Sleep Stage Estimation Method Based on Circadian Rhythm(RSSE-CR)

Problem of RSSE

The three graphs in Figure4 show the sleep stage estimation by R&K, the sleep stage estimation by RSSE and the baseline by the average of medium frequency component of the heart rate. In the graphs at the top and middle, the vertical axis indicates sleep stage, while the horizontal axis indicates time of sleeping. In the bottom one, the vertical axis indicates raw heart rates, while the horizontal axis indicates time of sleeping. The yellow line indicates the raw heart rates, the green line shows that medium frequency component of the heart rate and the black one describes the average

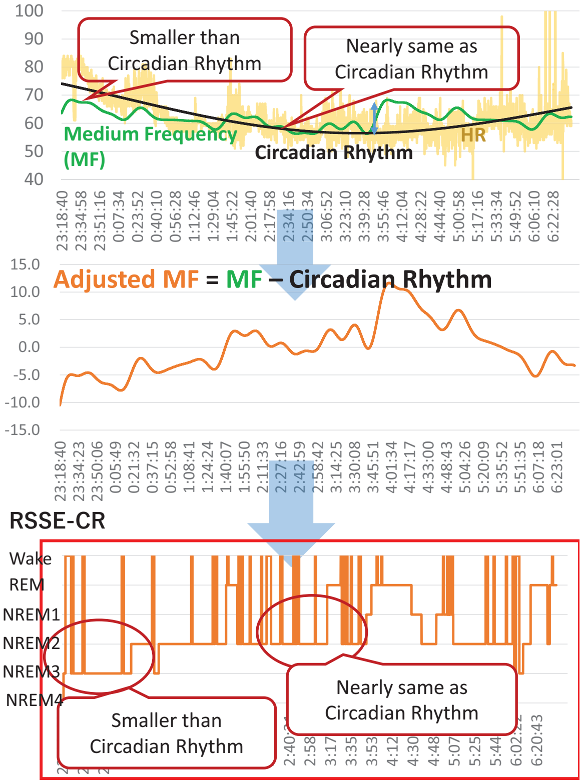


Figure 5: an examle to derive the sleep stage estimation in RSSE-CR

of medium frequency. RSSE estimates sleep stage by standardization depended on the baseline which is equal to the average of medium frequency component of heart rate as graph at the bottom in Figure4. Graph at the top in Figure4 shows the sleep stage estimation by R&K and the evaluations described are different from the sleep stage by RSSE. Focused on graphs at the middle and bottom one, the sleep stage described by the former remarkable circle is evaluated as NREM1 because the medium frequency is higher than the baseline. With the same way, the evaluation described by the latter remarkable circle is effected by the position of the medium frequency which is lower than the baseline. In general, human get to sleep deepest in the early time of sleeping and it becomes light toward wake, however, these tends could be related with personal life style. In RSSE, the baseline is fixed by the average of medium frequency, so that sleep rhythm is not taken into account. In other words, it is essential to consider the sleep depth variability with time elapsed for the high accuracy in sleep stage estimation. Thus, the proposed method attempts to take circadian rhythm account into for sleep stage estimation with higher accuracy.

Mechanism: The adjusted medium frequency

Figure5 shows the baseline by circadian rhythm and the relation with the medium frequency, the adjusted medium frequency and the sleep stage estimation by RSSE-CR from top to bottom. In the graph at the top, the vertical axis indicates raw heart rates. The yellow line indicates the raw heart rates, the green line shows the medium frequency component of the heart rate and the black one shows circadian rhythm. In the middle graph, the vertical axis indicates values in the standardization. In the bottom, the vertical axis indicates the sleep stage. In all graphs, the horizontal axis indicates time of sleeping. The adjusted medium frequency is obtained by a difference between the medium frequency component of the heart rate and the low frequency in order to take circadian rhythm into account as Figure5 shows. While the graph at the bottom in Figure4and at the top in Figure5 indicated same heart rate and the medium frequency, the change the baseline from the average of medium frequency to circadian rhythm made the difference as the view of the medium frequency. For instance, the sleep stage surrounded by the former remarkable circle at the graph in Figure5 was estimated as NREM3 because the medium frequency was lower than circadian rhythm as graph at the top shows. At the same way, the sleep stage highlighted by the latter circle at the bottom graph was estimated as NREM2 because the medium frequency was almost as same as circadian rhythm at the top graph. In this way, to pay attention to the relation with circadian rhythm adjust the sleep stage in comparison with RSSE.

Algorithm

In order to estimate sleep stage, RSSE-CR adds more two processes such as step(d) and step(e) in addition to RSSE. As overall processing flow, the following deals are executed in se- quence: (1) to obtain biological data from the mattress sen- sor; (2) to calculate both the medium frequency and the low frequency component of the heart rate based on the regres- sion of the trigonometric function(referred to (step a: note that step a is demonstrated in previous chapter) and (step d)); (3) to acquire the adjusted medium frequency; (4) to standard- ize and to let the adjusted medium frequency weighted with parameter (referred to in (step e)); (5) to discretize the ad- justed medium frequency(step f) and (6) to compensate the evaluation of Wake and REM.

(step d)Estimation of circadian rhythm RSSE-CR employs the low frequency component of heart rate as circadian rhythm, while circadian rhythm is generally obtained by measuring core body temperature(Goel et al. 2011). Because core body temperature is related with heart rate(Vandewalle et al. 2007) and also the low frequency with a period of twenty-four hours which is one of associated waves in heart rate is adequate to replace circadian rhythm also having a period of twenty-four hours. Based on this idea, RSSE-CR supposes that circadian rhythm is gained by a low frequency of a heart rate. Especially, in this paper, we use the low frequency of a heart rate in thirty-six hours circle as circadian rhythm. Specifically, it could be calculated by formula (1) (with following parameter: $N = 1$, $L = 2^{17} \approx 36$ hours). While circadian rhythm is twenty-four hours cycle, the reason why we employ the low frequency in thirty-six hours cycle as circadian rhythm is that the calculation of frequency component of heart rate in FFT is set by power of 2.

(step e)Calculation of the adjusted medium frequency To adjust a gradient of circadian rhythm, RSSE-CR executes the weighting deal to the adjusted medium frequency. The brackets $\{\}$ in Formula(8) indicates the calculation for the adjusted medium frequency as the graph at the middle in Figure5 shows, and $f(t)$ denotes the adjustment by parameter β which attenuates amplitude of the adjusted medium frequency. In this case, we set the range of parameter β to less than 1.0 in increments of 0.1 from 0.1 and the weighing is processed as following formula on condition that both ϕ in formula (1) for the medium frequency and the low frequency are given as ϕ_{MF} and ϕ_{LF} appropriately by the calculations for each. $f(t)$ is used for the calculation of formula(9).

$$f(t) = \beta\{h(t, \phi_{MF}) - h(t, \phi_{LF})\} \tag{8}$$

(step f)Discretization of the adjusted medium frequency For discretization of the adjusted medium frequency, RSSE-CR calculates it as following formula,

$$s(t) = \begin{cases} 5 & \lceil \frac{(f(t)-ave)}{stdev} + 2 \rceil > 5, \\ 0 & \lceil \frac{(f(t)-ave)}{stdev} + 2 \rceil < 0, \\ \lceil \frac{(f(t)-ave)}{stdev} + 2 \rceil & \text{otherwise.} \end{cases} \tag{9}$$

In this formula, both $stdev$ and ave are fixed as the parameter $\beta = 1.0$ regardless of the parameter β .

Experiment

Experimental Setting

In order to verify the effectiveness of RSSE-CR, we conducted the subject experiment toward three people (age from twenty to sixty, two men and a woman) in three days. Each subject has slept to give results by in two kinds of ways to evaluate sleep stage at the same day as following; (1) Alice PDx made by Philips for the evaluation by R&K method (2) the EMFit sensor for the evaluation by RSSE and RSSE-CR.

This paper conducted the following two experimental cases:

Case 1: sensitive analysis (parameter β)

Case 2 comparison of RSSE-CR with RSSE

Evaluation criteria

The sleep stage estimation with ALICE PDx by R&K method is set as the correct estimation and we compared with each evaluation accuracy calculated by both the previous and proposed method in four sleep stage evaluation. Because four sleep stage evaluation is also used in common, we firstly attempt to evaluate sleep stage roughly in four stage. Additionally, for the evaluation in the parameter β to adjust the weight of circadian rhythm, this paper compared the sleep stage estimation by RSSE-CR with changing the parameter β with the estimated accuracy by.

Result

Case1 : Sensitive Analysis

Table2 shows the estimation accuracy with the four stage evaluation in each parameter β, and the subject and the day of experiment are represented capital letter and the number in each (e.g. M_1; data of the subject M in the first day). The accuracy had risen up as the parameter β is close to 0.1. Especially, in less than 0.7, each accuracy in all experimental data was higher than RSSE, and it tended to get same value when the parameter β was getting down. While the value of accuracy was got as the same, waveforms in each parameter were different, which means they became different from the original medium frequency component of the heart rate steadily. Figure6 shows the change of waveforms with parameter β in the result of subject M (shown as M_3 in the table2). While the vertical axis indicates sleep stage, the horizontal axis indicates time of sleeping. The blue line shows the sleep stage estimation by R&K, the orange dotted line shows the sleep stage estimation by RSSE-CR and the orange line shows the adjusted medium frequency. While table2 indicates the accuracy in four sleep stage evaluation, Figure5 is set in six sleep stage evaluation for examination of the waveform variability in detail.

Table 2: Estimation accuracy with changing parameter β

	M_1	M_2	M_3	H_1	H_2	H_3	K_1	K_2	K_3	Ave
1	52.1%	54.9%	63.5%	64.6%	60.0%	65.6%	62.9%	62.3%	59.5%	60.6%
0.9	56.9%	56.9%	66.9%	70.5%	59.4%	69.6%	68.7%	65.2%	60.5%	63.8%
0.8	62.6%	64.3%	72.3%	73.5%	67.6%	71.1%	74.0%	69.3%	64.7%	68.8%
0.7	65.4%	66.1%	73.3%	76.7%	68.2%	71.5%	76.8%	77.3%	72.1%	71.9%
0.6	69.1%	67.5%	73.7%	80.6%	69.4%	72.5%	76.9%	77.3%	72.1%	73.2%
0.5	70.1%	68.8%	73.6%	80.7%	69.8%	74.3%	76.9%	77.3%	72.1%	73.7%
0.4	70.4%	69.3%	73.3%	80.7%	69.8%	74.3%	76.9%	77.3%	72.1%	73.8%
0.3	70.4%	69.7%	73.3%	80.7%	69.8%	74.3%	76.9%	77.3%	72.1%	73.8%
0.2	70.4%	69.7%	73.3%	80.7%	69.8%	74.3%	76.9%	77.3%	72.1%	73.8%
0.1	70.4%	69.7%	73.3%	80.7%	69.8%	74.3%	76.9%	77.3%	72.1%	73.8%
RSSE	60.5%	56.0%	58.6%	62.9%	64.4%	57.7%	58.3%	58.7%	68.1%	60.6%

Focused on the parameter β, we could look at their effect at the different views by the changes in amplitude of the adjusted medium frequency and the sleep stage. First, in the view of amplitude with the adjusted medium frequency, as parameter β gets smaller, it gives strong effect in that the amplitude is attenuated. In other words, the difference becomes larger in comparison with the non-adjusted medium frequency. On the other hands, in the view of sleep stage, the changes of sleep stage also get small as the parameter β becomes small because the adjusted medium frequency converges the average. For instance, in comparison between parameter $\beta = 1.0$ and $\beta = 0.6$ in Figure6, the classification came to be correct in the highlighted parts. However, focusing on the estimation in parameter $\beta = 0.2$, there was remarkable difference compared with the estimation by the R&K method. While the wrong estimations between NREM1 and NREM2 did not effect on the accuracy, because we adopted the four sleep stage estimation in that both are divided into Light sleep in this experiment. Paying attention to the wave of adjusted medium frequency, the amplitude variability of the medium frequency got to be almost steady and became different from the original medium frequency, which designated that parameter β gave too strong influences in the amplitude of the adjusted medium frequency.

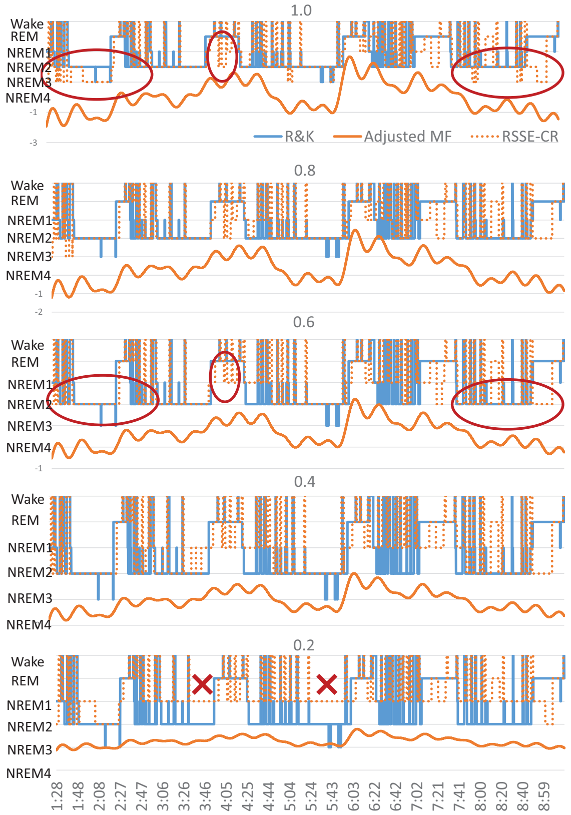


Figure 6: the waveform variability with parameter β

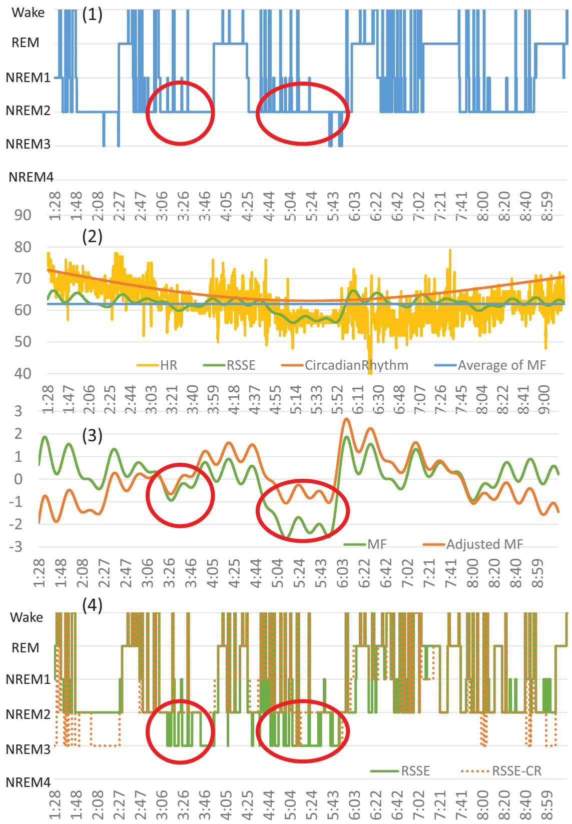


Figure 7: The waveform variability with the weighted circadian rhythm

Case2 : Comparison of RSSE-CR with RSSE

Figure7 shows four graphs in the result of subject M_3: (1) the sleep stage estimation by R&K; (2) the relation between the medium frequency and circadian rhythm; (3) the relation between the adjusted medium frequency(RSSE-CR) and the medium frequency(RSSE) and (4) the sleep stage estimation by both RSSE and RSSE-CR from top to bottom. While the vertical axes in four graphs of Figure7 indicate the sleep stage, heart rate, values of discretization and the sleep stage, the horizontal axes indicate time of sleeping in all graphs. The orange line in graph(2) shows the raw heart rate and the green line shows circadian rhythm. Also the green line indicates the medium frequency and the blue line is the average of medium frequency. In graph(3), the orange line indicates the adjusted medium frequency and the green one indicates the medium frequency. Finally, in graph(4), the green line indicates the sleep stage estimation by RSSE and the dotted orange line indicates the sleep stage estimation by RSSE-CR.

The estimated accuracy without parameter β, when the parameter $\beta = 1.0$, improved in five days out of in that nine days compared with RSSE. Focused on the former remarkable circle, the sleep stage estimated as NREM2 in RSSE-CR so that the value of the adjusted medium frequency was located between -1.0 and -2.0 in graph(3), which leads to the correct classification. However, the medium frequency was located lower than the adjusted one as the green line as graph(3) shows, which was caused the uncorrected evaluation as NREM3. At the same way, focused on the latter remarkable circle, RSSE-CR succeeded to estimate sleep stage correctly because the adjusted medium frequency was located between -1.0 and 0.0. All these results were affected by the adjusted medium frequency which depended on the relation with circadian rhythm.

On the other hand, with consideration for the weighting parameter β, when parameter $\beta = 0.6$, the estimated accuracy improved in nine days out of in that nine days in comparison with the estimation without the weight ing, which indicated that the weighting for circadian rhythm was effective in this experiment.

Discussion

From these results, it can be confirmed that RSSE-CR is able to estimate more accurate sleep stage than RSSE, however, it is necessary to reconsider the weight in waveform of circadian rhythm since we used a low frequency of thirty-six hours as circadian rhythm which is normal cycle in twenty-four hours. To adjust an appropriate waveform of circadian rhythm, this paper employed the adjusted medium frequency with parameter β. As the results of the weighted circadian rhythm, the parameter β is determined with the approximately 60 % discount of the raw circadian rhythm lest the parameter β effects too strong to sleep stage estimation such as showed by Figure6. Also, these results verified the possibility that it is suitable to employ a low frequency component of the heart rate as the replacement of the internal body temperature for circadian rhythm.

Conclusion

This paper suggested the sleep stage estimation method(RSSE-CR) to improve an estimated accuracy in addition to RSSE. To enhance the accuracy, we noticed circadian rhythm which governs the physical activity in animals since the previous methods in the sleep stage estimation did not take it account. Specifically, RSSE-CR regards the low frequency component of heart rate as circadian rhythm and calculates the adjusted medium frequency with revision of gradient of circadian rhythm in order to estimate sleep stage in four stage evaluation.

To guarantee the effectiveness of RSSE-CR, we measured its accuracy of four sleep stage estimation using 9 days of sleep data of three subjects and compared it with that of the Harada's method as the conventional method. As the experimental results, the following implications have been revealed: (1) the accuracy of the sleep stage estimation in 5 days out of that in 9 days were improved by RSSE-CR in comparison with RSSE; and (2) the parameter β (which determines the discount rate of curve of the circadian rhythm) should be set around 60%, meaning that a raw circadian rhythm (i.e., no discounted rhythm) affects strongly to the sleep stage while the highly discounted circadian rhythm (e.g. 30% discounted rhythm) does not contribute to accurately estimating the sleep stage.

As the future works, the following research must be done in the near future: (1) we should improve the classification criteria of the sleep stage because RSSE-CR classifies the sleep stage on the assumption of the standard normal distribution, which does not reflect the actual stage; and (2) we should also improve the accuracy of the sleep stage in six stage evaluation, which would be available to recognize a circumstance in sleeping more detail.

References

Ministry of Health, Labor and Welfare. 2015. National Health and Nutrition Survey http://www.mhlw.go.jp/file/04-Houdouhappyou-10904750-Kenkoukyoku-Gantaisakukenkouzoushinka/kekkagaiyou.pdf

Kamibayashi, Y., Hagiwara, K. 2012. An Approach on Estimation of Sleep Cycle Using Occurrence Rate of Body Movements. *Transactions of the Japan Society of Mechanical Engineers*, 50(1), 99-104. In Japanese.

Sasaki, N.,Kagawa, M,.Suzumura, K.,Matsui, T. 2015. Sleep Stage Estimation by Body Movement Index and Respiratory Interval Indices using Microwave Radars. *Transactions of the Japan Society of Mechanical Engineers*, 53(4), 209-216. In Japanese.

Komine, T., Takadama, K., Nishino, S. 2016. Toward the Next-Generation Sleep Monitoring/Evaluation by Human Body Vibration Analysis. *2016 AAAI Spring Symposium Series.*

Watanabe, T., Watanabe, K. 2002. Estimation of the Sleep Stages from the Bio-Data Non-invasively Measured in the Sleep *Transactions of the Society of Instrument and Control Engineers*, 38(7), 581-589. In Japanese.

Harada, T., and Takadama, K.2016. Real-Time Sleep Stage Estimation from Biological Data with Trigonometric Function Regression Model. *AAAI Spring Symposium: Wellbeing Computing: AI Meets Health and Happiness Science.*

Harada, T., and Takadama, K.2017.Improving Accuracy of Real-Time Sleep Stage Estimation by Considering Personal Sleep Feature and Rapid Change of Sleep Behavior. *AAAI Spring Symposium: Wellbeing Computing: AI Meets Health and Happiness Science*.

Goel, N., Van Dongen, H. P., & Dinges, D. F. (2011). Circadian rhythms in sleepiness, alertness, and performance. In Principles and Practice of Sleep Medicine (Fifth Edition) (pp. 445-455).

Vandewalle, G., Middleton, B., Rajaratnam, S. M., Stone, B. M., Thorleifsdottir, B., Arendt, J., & DIJK, D. J. (2007). Robust circadian rhythm in heart rate and its variability: influence of exogenous melatonin and photoperiod. Journal of sleep research, 16(2), 148-155.

Takadama, K., Hirose, K., Matsushima, H., Hattori, K., & Nakajima, N. (2010). Learning multiple band-pass filters for sleep stage estimation: towards care support for aged persons. IEICE transactions on communications, 93(4), 811-818.

Ensemble Heart Rate Extraction Method for Biological Data from Water Pressure Sensor

Fumito Uwano, Keiki Takadama
The University of Electro-Communications
1-5-1, Chofugaoka, Chofu-shi, Tokyo, Japan
{uwano@cas.lab, keiki@inf}.uec.ac.jp

Abstract

This paper proposes new heart rate estimation method for biological data from sleep monitor sensor toward estimating sleep stage accurately. Concretely, we employed two heart rate estimation methods, and integrated the two methods as weak estimator. One of the two methods calculates power spectrum from the biological data by FFT, and selects the frequency with maximum spectrum as heart rate (HR). The other calculates power spectrum as a same manner of the former method, and selects the frequency which indicates the half size of all power spectrum as HR. To validate the effectiveness of EHEM, this paper applies EHEM to pressure data from sleep monitor sensor. From the result, EHEM can extract HR accurately, and prevent from outliers generated by HEM-FFT. We are going to research (1) what method gives good influence to EHEM, and (2) how to integrate the HRs extracted from the methods.

Introduction

Lately, sleep is important topic for not only human helthcare but also human well-being. Especially, monitoring method for whether they are sleeping deeply or not is required and studied by many researcher. From this reason, several sleep monitoring methods were proposed. One of most standard sleep monitoring method is Rechtschaffen and Kales (R&K) method which calculates the sleep stage by the brain wave from the electroencephalogram (EEG), electromyogram (EMG), or electrooculography (EOG) (Rechtschaffen and Kales 1968). R&K method can estimate sleep stage accurately, but requires many kinds of data which is acquired from many sensors tester wearing. In addition, the sensors make stress to the tester, and prevent from sampling the data of the tester without the stress. To tackle this issue, sleep stage estimation methods based on heart rate data sampled from pressure sensor were mainly studied. Watanabe proposed the sleep stage estimation method based on the fluctuation of the heart rate obtained with a non-contact device (Watanabe and Watanabe 2004). This method estimates the sleep stage based on the intermediate frequency component of the heart rate measured by a non-contract device. The above things suggest that extracting the heart rate data accurately from the pressure sensor is important to estimate accurate sleep stage.

Pressure sensor measures pressure made by a human, *e.g.*, body moving, respiration, and heart beat. The pressure sensor acquires the data composed of these pressure data. Since respiration and heart beat occur with different cycle respectively, the heart rate is generally extracted by FFT from the data. Note that the cycle of the heart rate is changable for sympathetic and parasympathetic behaviors. However, these methods are weak for the data's uncertainty because they assume that the only one power spectrum can express all power spectra related to the heart rate (Tsuchiya et al. 2008). To solve this issue, we propose the heart rate extraction method with robustness against outliers, and propose new new method which integrates two heart rate extraction methods including the method based on FFT in order to surpress influence by the data's uncertainty.

This paper is organized as follows. We explain heart rate extraction technique in the section of Heart rate extraction, and the pressure sensor employed by us in the section of Water pressure sensor. Next, we introduce the proposed method in the section of Ensemble heart rate extraction method (EHEM), and explain and discuss experiment for effectiveness of the proposed method in the section of Experiment. At the end, we conclude this research.

Heart rate extraction

Heart rate (HR)

Heart rate (HR) indicates the number of heart beat for one minute. Generally, two kinds of data are measured as HR: blood pressure and body move. Heart rate measurement based on blood pressure is more accurate than that based on body move. However, the tester has to wear the sensor to measure the blood pressure ,*i.e.*, it is measured with stress. On the other hand, the measurement based on body move can acquire data without stress for the tester because the employed sensors (*e.g.*, pressure sensor and doppler sensor) do not require the tester's wearing. However, this measurement cannot acquire only HR. From this reason, heart rate extraction method must be required when you employ the measurement based on body move. This paper employ sleep monitor sensor as the pressure sensor, and the measurement based on body move.

Heart rate extraction method

Generally, HR is extracted based on FFT (Ttsuchiya et al. 2008) (called HEM-FFT in this paper). Figure 1 shows flow of heart rate extraction based on FFT. First, the method acquires certain range of pressure data and applies them to window function (process 1). Next, the method calculates power spectrum from this data by FFT (process 2), and selects frequency whose power spectrum is largest of all (process 3). These processes are continued until the unprocessed data do not exist. If they exist, process is returned to process 2 (process 4).

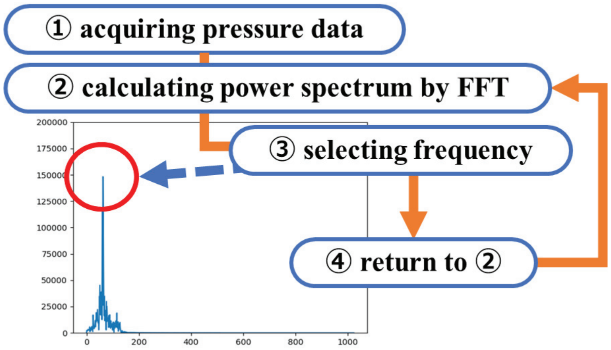


Figure 1: Flowchart

Water pressure sensor

Product specification

This paper employ TANITA sleep scan SL-511-WF as water pressure sensor. Figure 2 shows the image of the sleep monitor sensor. Table 1 shows the details of the sleep monitor sensor. The sensor is made from acrylonitrile butadiene styrene (ABS) and poly vinyl cloride (PVC). The sensor performs well with 7V, 1.7A, 5-35 degree celsius, 5-80 humidity. The sensor is generally called body move sensor, and class I medical device of number Q5B1X00001000003. The sensor measures pressure data which is happened when a human moves itself with 16Hz as sampling frequency and 0.5Hz and more as frequency band.

Figure 2: TANITA sleep scan (SL-511-WF)

Data format

The water pressure sensor can measure four pressure data in four sensors. In addition, this sensor can estimate the pressure data of the heart rate part from these pressure datas. This paper utilize these pressure data. Figure 3 shows the example of those pressure data. In this figure, vertical axis indicates magnitude of the pressure, while horizontal axis indicates time. The pressure value is from -2047 to 2048, and the unit of this horizontal axis is one 16th of a second.

Table 1: Sensor details

materials	ABS, PVC
output	7V, 1.7A
templeture range	5-35 degree celsius
humidity range	5-80%
general name	body move sensor
medical device class	class I
medical equipment number	Q5B1X00001000003
sampling frequency	16Hz
frequency band	0.5Hz and more

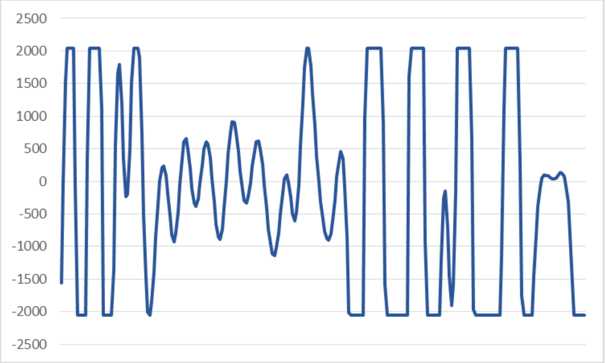


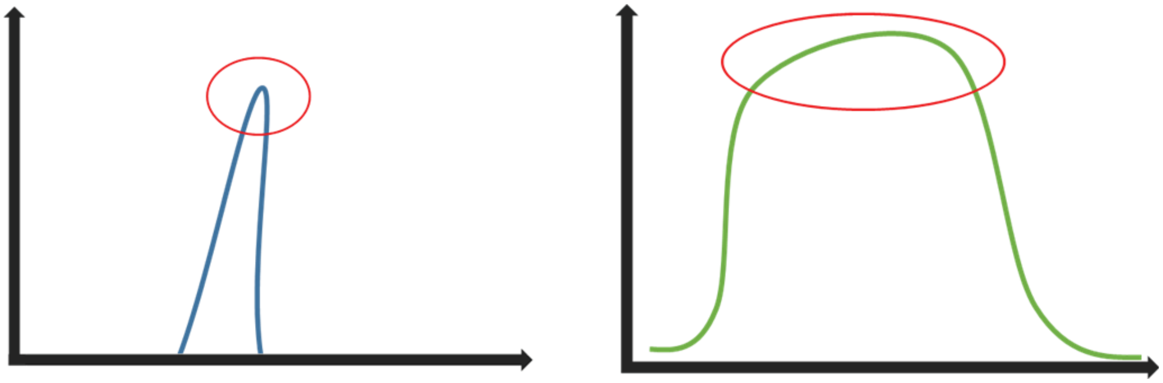
Figure 3: Example data

Ensemble heart rate extraction method (EHEM)

As machine learning techniques, there is ensemble learning. This generates one result by integrating several results acquired from several learning methods (or several learners based on one learning method and different data). This technique can prevent from error which one learning method generates because different methods seldom generate same error. This paper proposes ensemble heart rate extraction method (called EHEM) by utilizing this technique. Concretely, we proposes new heart rate extraction method called HEM-HS (we introduces HEM-HS in section below), and integrates HEM-FFT and HEM-HS to make EHEM.

Heart rate extraction based on half size (HEM-HS)

This paper proposes new method for one heart rate extraction method. This method has the same processes as HEM-FFT, but this method selects frequency which indicates the half size of all power spectra as HR (called HEM-HS). If the power spectra of several frequencies is almost same value. Since HEM-FFT estimates HR from the largest spectrum, if the power spectra of several frequencies are almost same



(a) One peak of spectra

(b) Several peaks of spectra

Figure 4: Two kinds of spactra

value, it is influenced easily from outliers. HEM-HS has robustness for the outlier because it estimates HR from the size of all power spectra. we explain the effectifeness of HEM-HS in two kinds of spectra empirically assumed below.

- One peak of spectra
  Figure 4a shows the spactra with one peak. HEM-FFT and HEM-HS estimate the frequency around this peak as HR.

- Several peaks of spectra
  Figure 4b shows the spactra with several peaks. HEM-FFT estimates the certain frequency among these peaks, while HEM-HS estimates the frequency of the center of these peaks as HR.

In the situations of one peak and two peaks, both methods can estimate HR accurately, but HR of HEM-FFT is more accurately than HR of HEM-HS. In the situation of several peaks, HEM-FFT cannot estimate HR accurately, while HEM-HS can estimate HR accurately. we integrates both method in order to utilize strong points and decrease weak points. The integration way is explained below.

Ensemble policy

As machine learning techniques, there is ensemble learning. This generates one result by integrating several results acquired from several learning methods (or several learners based on one learning method and different data). This technique can prevent from error which one learning method generates because different methods seldom generate same error. We proposes ensemble heart rate extraction method (called EHEM) by utilizing this technique. As for the concretely method in this paper, EHEM extracts HRs by HEM-FFT and HEM-HS, and compares the two HRs. If difference between the two HRs is over certain threshold, this method employs the HR by HEM-HS; otherwise, this method employs the HR by HEM-FFT. This threshold indicates priority of whether EHEM accepts the HR of HEM-FFT. If the threshold is large, EHEM accepts HR of HEM-FFT in a lot of time.

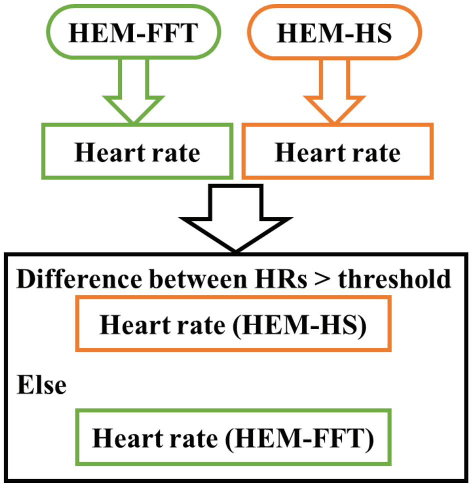


Figure 5: EHEM based on HEM-FFT and HEM-HS

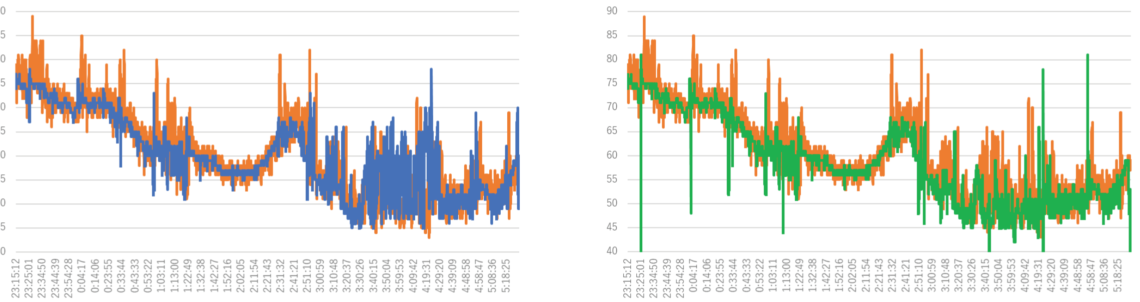
Experiment

Experimental details

This paper utilizes pressure data from water pressure sensor while tester sleeping, and compares the HRs extracted by EHEM and HEM-FFT, and measured by TEIJIN as true HR. This paper evaluates the behavior of the HR. Concretely, this paper evalueates whether the extracted HR bahaves along to the HR of TEIJIN. In additon, average, standard deviation, and max value of the errors among the HRs of EHEM, HEM-FFT, and TEIJIN are evaluated, respectively. We utilize 26 kinds of pressure data from 12 males or females people, and ages of them are from 20s to 70s. The above threshold is 10 in this experiment.

Results

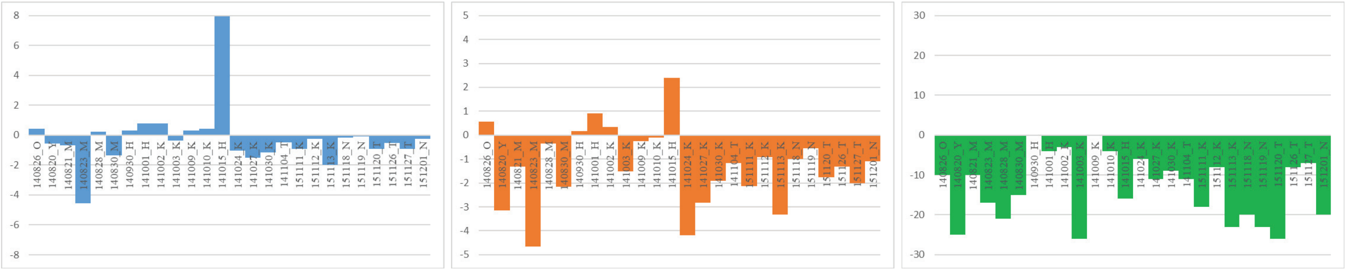
Figures 6 shows the behavior of the HR by EHEM and HEM-FFT in one example. In this figure, blue, green, and orange lines show the HR by EHEM and HEM-FFT, and that measured by TEIJIN, respectively. From this result, HRs



(a) Result of EHEM

(b) Result of HEM-FFT

Figure 6: Result (blue: EHEM, green: HEM-FFT, orange: TEIJIN)



(a) Average of errors

(b) Standard deviation of errors

(c) Max value of errors

Figure 7: Average, standard deviation, and max value of errors in all examples

of EHEM and HEM-FFT can behave along to the HR of TEIJIN. The HR of HEM-FFT sometimes becomes outlier, while that of EHEM can prevent from becoming outlier. Table 2 shows average, standard deviation, and max value of the errors of HR between EHEM and TEIJIN (upper side), HEM-FFT and TEIJIN (lower side) respectively. The HR of EHEM is smaller than that of HEM-FFT in all attributes. Figure 7 is the differences of the errors between EHEM and HEM-FFT in each data. Left side, middle side, and right side of Fig. 7 indicate the differences of the average, the standard deviation, and the max value of the errors, respectively. Vertical and horizontal axes are the differences and kinds of data, respectively. In this figure, if the bar is negaitive, EHEM performs better than HEM-FFT, otherwise, HEM-FFT performs better than EHEM. From this result, EHEM performs better than HEM-FFT in almost whole case of the data, though it performs worse than HEM-FFT in one data “141015_H” in terms of the above three points.

Table 2: Difference between EHEM and HEM-FFT

method	average	standard deviation	max value
EHEM	4.89	5.38	39.8
HEM-FFT	5.13	6.66	52.9

Discussion

From the results, EHEM can extract HR accurately. From Table 2 and Fig. 7, since the HR extracted by EHEM has small error than that extracted by HEM-FFT, EHEM can prevent the error made by HEM-FFT, and extract HR along to the true HR accurately. That is because EHEM is based on HEM-HS and HEM-FFT. HEM-HS can extract HR without outlier, though the accuracy of HEM-HS is not better than that of HEM-FFT in each time. From the result, it is clear that EHEM can utilize the effectiveness of HEM-HS and the accuracy of HEM-FFT. We discuss the performance of EHEM in terms of each point of view below.

Result of HEM-HS

Figure 9 is the result of HEM-HS in one example. Vertical and horizontal axes indicate the extracted HR and time, respectively. Violet line is the result of HEM-HS, while orange line is HR of TEIJIN. In this figure, HEM-HS can extract HR without outlier, but cannot capture the shape of true HR more accurately than HEM-FFT. That is because HEM-HS utilize half size of all spectra for extraction. HEM-HS can extract HR from the shape of the spectra by utilizing size of them. This way has robustness to outlier. However, HEM-HS cannot capture the small difference of pressure by the HR changing because the spectra are the similar shapes with each other in each time. As for the result, the HR by HEM-HS has smaller amplitude than other HRs.

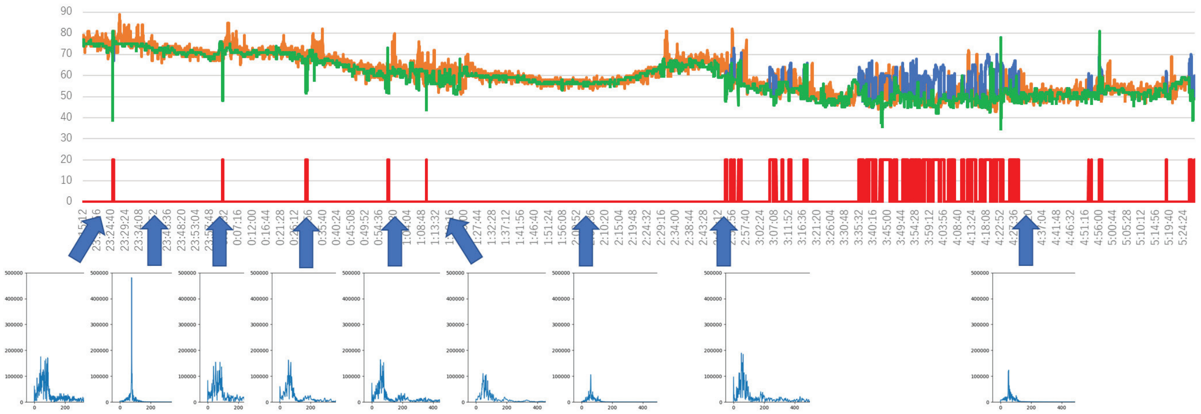


Figure 8: Details of EHEM

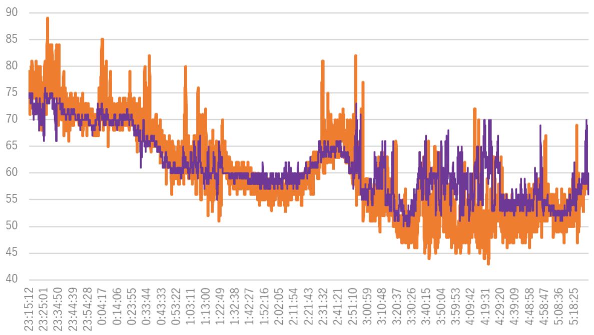


Figure 9: Result of HEM-HS

Candidate of EHEM

Figure 8 shows the detail of EHEM performance in one example of the experiment. Lines in upper side of this figure show HRs of EHEM, HEM-FFT, and TEIJIN, respectively. A line in middle side of this figure shows which method EHEM utilizes. EHEM has utilized HEM-HS while this line is in up side, and has utilized HEM-FFT while it is in low side. Figures in lower side of this figure show spectra in certain time. The arrow from each spectrum indicates to the time when it is occured. In Fig. 8, EHEM rejects HR of HEM-FFT and accepts HR of HEM-HS, whenever HEM-FFT extracts HR with extreme deviation from true HR. In addition, there are two types of the spectra in Fig. 8, the spectrum with one peak and several peaks, respectively. HEM-FFT and HEM-HS perform well in the former and the latter spectra, respectively. EHEM can select appropriate method along to the above fact.

Difference among sensors

In this paper, we utilize pressure data by heart beat extracted from the water pressure sensor. Figure 10 shows the extracted HR by EHEM from raw data measured by four sensors in one example. The extracted HR by EHEM from the extracted data is shown in Fig. 6a. Vertical and horizontal axes indicate HR and time, respectively. Blue line is HR by EHEM, while orange line is true HR measured by TEIJIN. From these results, the extraction from the raw data is difficult for EHEM, preprocessing is required. However, the HRs in Fig. 10c and 10d have behaved along to true HR in first half, while the HRs in Fig. 10a and 10b have behaved along to true HR in latter half. On the other hand, Table 3 shows the average, the standard deviation, and the max value of the errors. From this table, the results based on data from each sensor are worse than that of extracted pressure data, but averagely the results are less than almost 6.

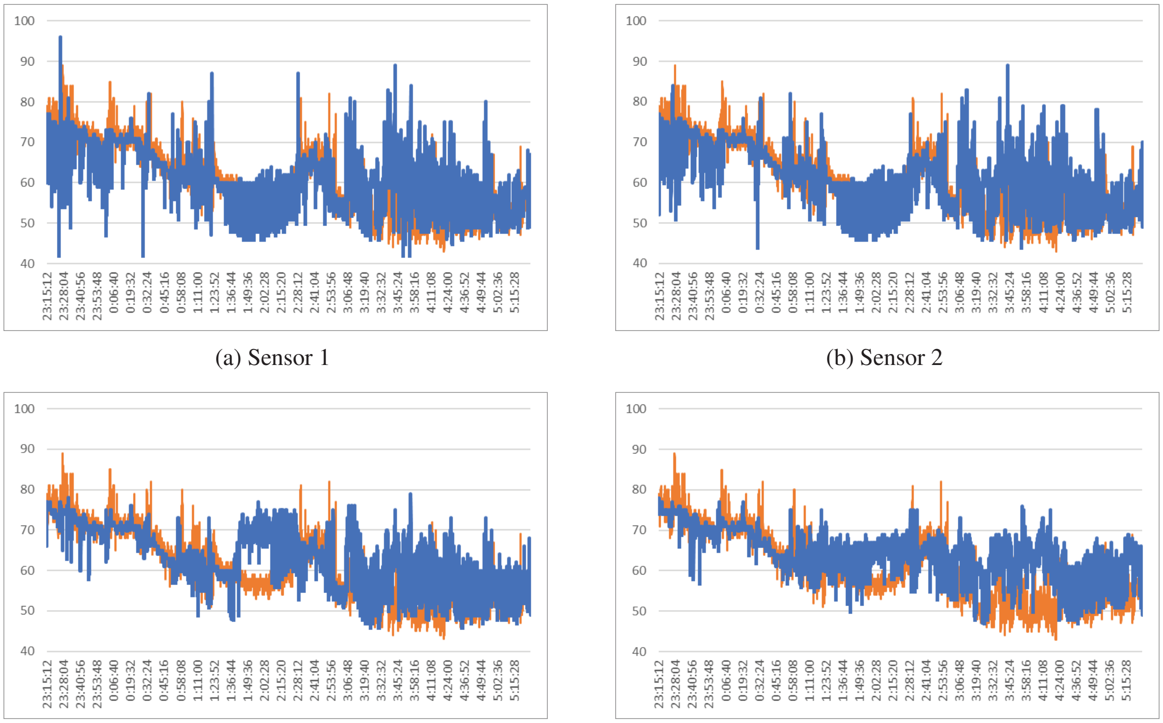
Table 3: Difference among each sensor in EHEM

sensors	average	standard deviation	max
heart rate	3.29	3.72	26
sensor 1	5.18	5.23	39
sensor 2	5.16	5.19	39
sensor 3	5.55	5.09	29
sensor 4	6.02	5.06	27

Table 4: Details of Tab. 3

	sensors	average	standard deviation	max
first	sensor 1	3.79	4.39	31
	sensor 2	3.35	3.68	27
	sensor 3	2.62	2.70	24
	sensor 4	2.53	2.50	19
middle	sensor 1	4.58	4.43	34
	sensor 2	4.63	4.45	29
	sensor 3	6.97	5.33	24
	sensor 4	5.83	3.80	24
last	sensor 1	7.19	6.10	39
	sensor 2	7.54	6.21	39
	sensor 3	7.06	5.38	29
	sensor 4	9.72	5.54	27

Table 4 shows the detail of Tab. 3. This table show the difference among each sensor in three periods. Upper side



(a) Sensor 1

(b) Sensor 2

(c) Sensor 3

(d) Sensor 4

Figure 10: Result of each sensor

is that in first one third time, middle side is that in next one third time, bottom side is that in last one third time. From this table, the data of sensors 3 and 4 behave like the pressure for HR in first period, the data of sensors 1 and 2 behave like the pressure for HR in middle period. However, nothing behaves like the pressure for HR in last period. Therefore, EHEM might perform well by utilizing raw data, but these results suggests the requirement for data using.

Conclusion

This paper proposes ensemble heart rate extraction method called EHEM which integrates two heart rate extraction methods (HEM-FFT and HEM-HS) in order to decrease errors generated by HEM-FFT, and improve accuracy of heart rate extraction method. Concretely, this paper employs HEM-FFT which regards frequency with maximum power spectrum calculated by FFT as HR and HEM-HS which regards frequency that indicates the half size of all power spectrum as HR, and EHEM regards the HR of HEM-HS as true HR when difference between the two HRs is over 10; otherwise EHEM regards the HR of HEM-HS as true HR. To validate the effectiveness of EHEM, this paper applies EHEM to pressure data from sleep monitor sensor. From the result, EHEM can extract HR accurately, and prevent from outliers generated by HEM-FFT.

We are going to research (1) what method gives good influence to EHEM, (2) how to integrate the HRs extracted from the methods, and (3) how to utilize raw data for heart rate extraction.

References

Rechtschaffen, A., and Kales, A. 1968. A manual of standardized terminology, techniques and scoring system for sleep stages of human subjects.

Tsuchiya, N.; Yamamoto, K.; Nakajima, H.; and Hata, Y. 2008. A comparative study of heart rate estimation via air pressure sensor. In *2008 IEEE International Conference on Systems, Man and Cybernetics*, 3077–3082.

Watanabe, T., and Watanabe, K. 2004. Noncontact method for sleep stage estimation. *IEEE Transactions on Biomedical Engineering* 51(10):1735–1748.

Does Digital Dementia Exist?

Hideya Yamamoto, Kaoru Ito, Chihiro Honda, Eiji Aramaki
Nara Institute of Science and Technology
{yamamoto.hideya.xx7, kito, chonda, aramaki}@is.naist.jp

Abstract

In recent years, various mobile communication devices such as smartphones are becoming increasingly popular. Because of the convenience brought by such devices, they are apparently becoming indispensable. However, several studies have indicated that these devices have detrimental effects on our cognitive abilities; some studies have described this phenomenon as digital dementia (DD). Media multitasking and the heavy use of mobile devices are suggested as some factors causing DD. Nevertheless, evidence linking the overuse of mobile devices and DD remains scarce. This study was conducted to elucidate the existence and possible causes of DD using crowdsourcing, which facilitates recruitment of numerous study participants. We investigate the usage of information devices and cognitive ability. Via crowdsourcing, one thousand study participants were recruited. Results suggest that the age when one begins using mobile devices as well as the heavy usage of those are potential factors leading to cognitive decline. We want to sound the alarm on the use of mobile terminals, which might cause severe disorder.

Introduction

With rapid innovation in the information and communications technology (ICT) area, our lifestyles have been changing dramatically. Sometimes such changes cause new disorders, such as smartphone related disorders. In recent years, mobile communication devices, including smartphones and tablets, have rapidly become popular, enabling virtual communication via internet. Because of that convenience, the devices are becoming increasingly indispensable and addictive.

However, use of those devices, particularly overuse, has been emphasized as a cause for alarm. Actually, not a few studies have found possible adverse effects of overuse: multitasking has a relation to a decline of attention (Ralph et al. 2014) and memory (Ophir, Nass, and Wagner 2009); overuse of smartphones is related to a declining tendency of thought (Barr et al. 2015). Several reports of the literature have described these symptoms as digital dementia (DD) because of their symptoms, which are analogous to those of Alzheimer's disease. Several reports have described that DD derives from two factors: (1) overuse of information devices such as smartphones and (2) daily multitasking habits. Nevertheless, no reliable evidence exists for DD because of the difficulty of designing valid experiments. Moreover, counter studies of DD have found no relation between media multitasking and cognitive ability (Ralph et al. 2015). Therefore, even today, DD is a vague and controversial concept. This situation naturally motivates research to explore the existence of DD and its putative mechanisms. Although recruiting participants using crowdsourcing generally bottlenecks unbiased recruitments because almost all crowdworkers are heavy users of digital devices, we intentionally used crowdsourcing in expectation of the biases. For this study, heavy users of digital devices are suitable as participants. From analyzing the data of recruited 1000 participants, existence of DD was apparent. Moreover, new potential factors causing DD were found.

Method

We used Yahoo! Crowdsourcing[1] to investigate the state of each participant (Fig. 1) using a questionnaire.

Recruitment Crowdsourcing was used for recruitment for two reasons: (1) people who use those devices to some extent in daily life are suitable as participants; (2) it is easy to recruit many participants.

Questionnaire We investigated the usage of information devices. Information devices were classified into two categories: (a) mobile terminals and (b) computers. Usage was classified into three categories: (1) age at start of use (start age), (2) usage time per day, (3) time for private use per day. Cognitive ability and mental test scores were also calculated from the responses to the questionnaires. Calculated features were mindfulness (Brown and Ryan 2003), memory (Kazui et al. 2003), conscientiousness (Wada 1996), happiness [2], and extraversion (Wada 1996), and stress (Imazu et al. 2006).

Results

As a result, 1000 participants (male 482, female 517, other 1) were recruited via crowdsourcing during September–October 2017. The results are explained below.

[1] https://crowdsourcing.yahoo.co.jp/

[2] http://www.med.oita-u.ac.jp/oita-lcde/WHO-5[1].pdf

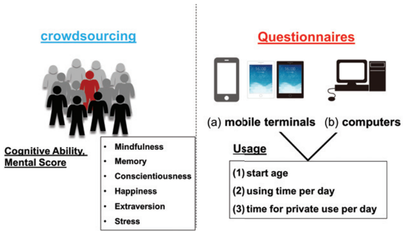


Figure 1: Recruitment and questionnaire.

(a) (1) Start age: Significant variation was found in Mindfulness, Memory, and Conscientiousness (one-way ANOVA, $p < 0.01$). (2) Usage time per day: Significant variation was found for Mindfulness, Conscientiousness, Extraversion, and Stress. (3) Time for private use per day: Significant variation was found only for Mindfulness. (Table 1)
(b) (1)–(3) In no usage category was significant variation found among scores.

	start age	using time	time for private use
Mindfulness	✓	✓	✓
Memory	✓		
Conscientiousness	✓	✓	
Happiness			
Extraversion		✓	
Stress		✓	

Table 1: Bariation between mobile terminal groups (one-way ANOVA, ✓: $p < 0.01$).

Additionally, we classified participants into two groups according to start age: under 20 years old (Early Start) and 21 plus (Late Start). Comparison of these two groups revealed that the Mindfulness score (higher score = less attentive) of the under 20 years old group was greater than that of the 21 plus group (45.29 ± 12.14 vs. 42.21 ± 12.25, $p < 0.01$). Moreover, the Memory score (higher score = less memory ability in daily life) was greater (12.01 ± 6.54 vs. 9.93 ± 5.81, $p < 0.01$). The stress score (higher score = more stressed) was also higher (7.05 ± 3.14 vs. 6.63 ± 3.10, $p < 0.01$). The Conscientiousness score (the score is high, more conscientious) was lower (46.38 ± 6.77 vs. 48.57 ± 7.21, $p < 0.01$) (Table 2).

	Early Start (n = 406)	Late Start (n = 594)	P-value
Mindfulness	45.29 ± 12.14	42.21 ± 12.25	< 0.01
Memory	12.01 ± 6.54	9.93 ± 5.81	< 0.01
Conscientiousness	46.38 ± 6.77	48.57 ± 7.21	< 0.01
Happiness	11.71 ± 5.05	11.69 ± 5.33	0.41
Extraversion	48.72 ± 4.99	49.42 ± 4.66	0.01
Stress	7.05 ± 3.14	6.63 ± 3.10	< 0.01

Table 2: Comparison of mental test scores between two groups related to the start age of mobile terminal use.

Discussion

The DD symptoms found in earlier studies appeared among participants recruited via crowdsourcing. Some examples are that the relation between the mobile usage per day and the Mindfulness score support this view. Moreover, results suggest that the starting age of mobile device use is also an important factor related to DD: the earlier a person starts using mobile terminals, the less attention and memory ability a person has and the greater the amount of stress felt.

Considering that the number of young people is dependent on smartphones in daily life, this trend can be expected to persist into the next decade. Consequently, during the next decade, DD might be regarded as a common state of human beings because many people around world are expected to be affected by DD caused by overuse of mobile devices. Follow-up investigations with appropriate control groups would be difficult to conduct.

Considering the points raised above, future works are the following: (1) investigating the relation between the hippocampus size and device-use starting age and (2) investigating the severity of DD's spread through smartphone applications, investigating applications that can notify users of overuse, etc. Although DD is not currently regarded as a severe problem, it must be monitored closely in the near future.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers JP16H06395 and JP16H06399.

References

Barr, N.; Pennycook, G.; Stolz, J. A.; and Fugelsang, J. A. 2015. The brain in your pocket: Evidence that smartphones are used to supplant thinking. *Computers in Human Behavior* 48:473–480.

Brown, K. W., and Ryan, R. M. 2003. The benefits of being present: mindfulness and its role in psychological well-being. *Journal of personality and social psychology* 84(4):822.

Imazu, Y.; Murakami, M.; Kobayashi, M.; Matsuno, T.; Shiihara, Y.; Ishihara, K.; Joh, Y.; and Kodama, M. 2006. Public health research foundation sutoresu chekku-risuto shouto foumu no sakusei/shinraisei/datousei no kentou (building and examination of the check list for stress). *Shinshinigaku (Psychosomatic Medicine)* 46(4):301–308.

Kazui, H.; Watamori, T.; Honda, R.; and Mori, E. 2003. Nihonban nichijyou chekku-risuto no yuuyousei no kentou (examination of usefulness of japanese daily check list). *Brain and Nerve Nou to shinkei (Brain and Nerve)* 55(4):317–325.

Ophir, E.; Nass, C.; and Wagner, A. D. 2009. Cognitive control in media multitaskers. *Proceedings of the National Academy of Sciences* 106(37):15583–15587.

Ralph, B. C.; Thomson, D. R.; Cheyne, J. A.; and Smilek, D. 2014. Media multitasking and failures of attention in everyday life. *Psychological research* 78(5):661–669.

Ralph, B. C.; Thomson, D. R.; Seli, P.; Carriere, J. S.; and Smilek, D. 2015. Media multitasking and behavioral measures of sustained attention. *Attention, Perception, & Psychophysics* 77(2):390–401.

Wada, S. 1996. Seikakutokuseiyougo wo mochiita biggu faibu syakudo no sakusei (building big five inventory with characteristical vocabulary). *Shinrigakukenkyuu (Psychology research)* 67(1):61–67.

Data Efficient Reinforcement Learning

Efficient Exploration for Constrained MDPs

Majid Alkaee Taleghan, Thomas G. Dietterich

School of Electrical Engineering and Computer Science
Oregon State University
Corvallis, OR 97331
alkaee,tgd@oregonstate.edu

Abstract

Given a Markov Decision Process (MDP) defined by a simulator, a designated starting state s_0, and a downside risk constraint defined as the probability of reaching catastrophic states, our goal is to find a stationary deterministic policy π that with probability $1 - \delta$ achieves a value $V^{\pi}(s_0)$ that is within ϵ of the value of the optimal stationary deterministic ν-feasible policy, $V^{*}(s_0)$, while economizing on the number of calls to the simulator. This paper presents the first **PAC-Safe-RL** algorithm for this purpose. The algorithm extends PAC-RL algorithms for efficient exploration while providing guarantees that the downside constraint is satisfied. Experiments comparing our CONSTRAINEDDDV algorithm to baselines show substantial reductions in the number of simulator calls required to find a feasible policy.

Introduction

This work is inspired by problems in natural resource management centered on the challenge of invasive species (Dietterich, Alkaee Taleghan, and Crowley, 2013; Taleghan et al., 2015). Computing optimal management policies for ecosystems is challenging because they exhibit complex spatio-temporal interactions at multiple scales. Many ecosystem management problems can be formulated as MDP (Markov Decision Process) planning problems (Sheldon et al., 2010). In a simulator-defined MDP, the Markovian dynamics and rewards are provided by a simulator from which samples can be drawn. Simulators in natural resource management can be very expensive to execute, so that the time required to solve such MDPs is dominated by the number of calls to the simulator.

Efficient MDP planning algorithms attempt to minimize the number of simulator calls before terminating and outputting a policy that is approximately optimal with high probability (Dietterich, Alkaee Taleghan, and Crowley, 2013). For unconstrained MDPs, the standard formulation of this is the notion of PAC-RL, first introduced by Fiechter (1994). This is in contrast to the PAC-MDP formalization, which minimizes various measures of infinite-horizon regret (Strehl and Littman, 2008). A common component of PAC-RL algorithms is to compute confidence intervals and explore using the optimism principle.

In many practical scenarios, such as natural resource management, a desirable policy needs to satisfy certain constraints imposed by decision makers. In these scenarios, maximizing the expected reward does not necessarily avoid rare catastrophic or dangerous situations. For example, in conservation problems, catastrophic outcomes include species extinction, long-term establishment of an invasive species, and severe wildfires. A standard approach to finding policies that avoid catastrophic states is to assign a large negative reward to those states (García and Fernández, 2015; Geibel and Wysotzki, 2005). This is equivalent to a so-called Big M method for establishing a lexicographic preference for policies that do not enter catastrophic states. However, this approach does not quantify the risk (probability) of entering a catastrophic state, nor does it determine whether there are policies that control this risk. A better approach is to adopt the Constrained MDP (C-MDP) formalism (Altman, 1999), which seeks to maximize one objective (e.g., economic value) while satisfying one or more constraints probabilistically. For example, in invasive species management, we can define a C-MDP to minimize the economic cost of invasive species management while ensuring that the probability of native species extinction is less than a specified threshold.

Recently, Geibel and Wysotzki (2005) developed a model-free Q-learning algorithm for C-MDPs. Their formulation is applicable to episodic tasks with a combination of absorbing catastrophic and goal states. As Geramifard (2012) pointed out, the Geibel, et al., work does not provide a performance guarantee on the result.

An alternative to constrained MDPs is to consider risk-sensitive objectives such as variance penalties, value at risk (VaR), and conditional value at risk (CVaR) (García and Fernández, 2015; Altman, 1999). Var and CVar optimize the α-quantile of the expected return, and CVaR has favorable mathematical properties. While these are all very interesting approaches, we find the constrained MDP formulation easier to understand and explain to stakeholders, and for this reason, we focus our efforts on C-MDPs.

A drawback of C-MDPs is that the optimal policy can be stochastic in some cases. Specifically, if there are c constraints, then the optimal policy may be stochastic in up to c states. From the perspective of our stakeholders, this stochastic behavior is confusing and undesirable. Hence, in this paper, we aim to find a stationary deterministic policy that satisfies a downside risk constraint as well as maximizing the discounted reward. We seek to do this while economizing on the number of calls to the simulator and while providing PAC guarantees both that the constraints are satisfied and that the resulting policy is within a fixed bound of optimality. This provides the first PAC-RL algorithm for deterministic policies in C-MDPs.

The paper is organized as follows. Section 2 introduces our notation for MDPs, C-MDPs, and confidence intervals. Section 3 introduces our new planning algorithm CONSTRAINEDDDV. Section 4 provides theoretical results. Section 5 presents an experimental evaluation of CONSTRAINEDDDV and a comparison with other methods. Section 6 concludes the paper. We evaluate our algorithms on an invasive species problem as well as on standard reinforcement learning benchmarks.

Problem Definition and Notation

Let a simulator-defined MDP consist of a start state s_0, a set of possible states S, a set of possible actions A, a discount factor $\gamma \in (0, 1]$ and a stochastic function F that maps from an input state-action pair (s, a) to a resulting state s' and reward r, where $s' \sim P(s'|s, a)$ is sampled according to the (unknown) transition function, $r \sim R(r|s, a)$ is sampled according to the unknown reward function, and $0 \leq r \leq R_{max}$. In this paper, we will assume that the reward is deterministic; our methods can be easily extended to handle stochastic rewards. A (deterministic) policy π is a function mapping from states s to actions $a = \pi(s)$. The value of the policy in the start state, $V^\pi(s_0)$, is the expected discounted cumulative reward:

$$V^\pi(s_0) = E\left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s = s_0\right].$$

Let $V_{max} = \frac{R_{max}}{1-\gamma}$ be the maximum possible value of any state under any policy. The corresponding minimum possible value is zero.

An optimal policy π^* maximizes $V^\pi(s_0)$, and the corresponding value is denoted by $V^*(s_0)$. The action-value of state s and action a under policy π is defined as $Q^\pi(s, a) = R(s, a) + \gamma \sum_{s'} P(s'|s, a)V^\pi(s')$. The optimal action-value is denoted $Q^*(s, a)$. Later, we indicate these functions with subscript R to distinguish them from the catastrophe value function.

Definition 1 *The occupancy measure μ of an MDP under policy π is defined as*

$$\mu^\pi(s) = \mathbb{E}_P\left[\sum_{t=0}^{\infty} \gamma^t I[s_t = s]|s_0, \pi\right],$$

where $I[\cdot]$ is the indicator function and the expectation is taken with respect to the transition distribution.

This is the cumulative discounted probability that the MDP will occupy state s under policy π for discount factor γ. It can be computed via dynamic programming on the Bellman flow equation (Syed, Bowling, and Schapire, 2008):

$$\mu^\pi(s) = I[s = s_0] + \gamma \sum_{s^-} \mu(s^-)P(s|s^-, \pi(s^-)). \quad (1)$$

This says that the discounted probability of visiting state s is equal to the sum of the probability that s is the starting state and the probability of reaching s by first visiting state s^- and then executing an action that leads to state s.

It is easy to show that

$$V^\pi(s_0) = \sum_s \mu^\pi(s)R(s, \pi(s)). \quad (2)$$

We adopt $\mu^{\pi^{UCB}}$ (also written as μ^{UCB}) as the occupancy measure computed based on the principle of optimism under uncertainty and maximum likelihood estimates of transition probabilities.

Let a subset of states $S_C \subset S$ be "catastrophic" states in the sense that we want to limit the probability of entering those states. Let us assume that all states in S_C are absorbing.

Definition 2 *For a policy π, the risk in state s is defined as*

$$\xi^\pi(s) = \sum_t \gamma_C^t P(s_t \in S_C|s, \pi), \quad (3)$$

which is the (discounted) probability of entering a catastrophic state when following π. γ_C denotes the catastrophe discount factor.

As a learning algorithm explores the MDP, it collects the following statistics. Let $N(s, a)$ be the number of times state-action pair (s, a) is simulated during learning and $N(s) = \sum_a N(s, a)$. Let $N(s, a, s')$ be the corresponding number of times that s' has been observed as the resulting state. Let $R(s, a)$ be the observed reward. Let $\hat{P}(s'|s, a) = N(s, a, s')/N(s, a)$ be the maximum likelihood estimate for $P(s'|s, a)$.

A $1 - \delta$ confidence interval is a pair of random variables $\underline{V}(s_0), \overline{V}(s_0)$ such that with probability $1 - \delta$, $\underline{V}(s_0) \leq V^\pi(s_0) \leq \overline{V}(s_0)$. Similarly, $\underline{Q}(s, a)$ and $\overline{Q}(s, a)$ denote the confidence bounds over the action-value functions. We follow the "Optimism Under Uncertainty" principle, and denote by π^{UCB} the policy based on an upper confidence bound on the action-value function, $\pi^{UCB}(s) = \arg\max_a \overline{Q}(s, a)$.

Definition 3 *(Fiechter, 1994). A learning algorithm is PAC-RL if for any discounted MDP $(S, A, P, R, \gamma, P_0)$, $\epsilon > 0$, $1 > \delta > 0$, and $0 \leq \gamma < 1$, the algorithm halts and outputs a policy π such that*

$$\mathbb{P}[|V^*(s_0) - V^\pi(s_0)| \leq \epsilon] \geq 1 - \delta,$$

in time polynomial in $|S|$, $|A|$, $1/\epsilon$, $1/\delta$, $1/(1 - \gamma)$, and R_{max}.

Optimal Policies for C-MDPs

Before delving into additional definitions for C-MDPs, let's clarify the class of optimal policies for C-MDPs. It has been shown that, unlike unconstrained MDPs, the optimal policies in C-MDPs are not necessarily stationary and deterministic and may depend on the starting state (Feinberg and Shwartz, 1996; Zadorojniy, Even, and Shwartz, 2009). In standard discounted unconstrained MDPs, one can find optimal policies that are stationary and deterministic from any state in $O\left(|S|^2|A|\right)$. In a C-MDP with two objectives (the standard value function and the risk of catastrophe), if the two objectives have unequal discount factors, then finding deterministic and stationary policies is NP-complete (Dolgov and Durfee, 2005; Feinberg, 2000; Chang, 2016). Optimal policies in C-MDPs with equal discount factors are randomized and stationary for a fixed starting state. The solution can be found by solving a linear program, where the dual variables represent the state occupancy measure, if the model is known. In our case where we only have one constraint, the optimal randomized policy is called a "1-randomized" policy (Zadorojniy, Even, and Shwartz, 2009). This means the difference between deterministic and the 1-randomized policy will arise in at most one state, where the randomized policy may choose probabilistically between two actions (Feinberg and Rothblum, 2012).

In this paper, we focus on finding a best policy in the class of stationary and deterministic policies with performance guarantees, even when a randomized policy is the optimal policy. It is a challenge to present a randomized policy to stakeholders. Feinberg (2008) points out that implementation of randomized policies is not natural in many applications, and the use of randomization procedures could increase the variance of the expected return. Boutilier and Lu (2016) also give an example of how randomized policy could be undesirable.

Additional Definitions for C-MDPs

Let Π be the space of deterministic polices over the constrained MDP $\mathcal{M}(\tau) = \langle S, A, P, R_R, R_C, \tau, \gamma, s_0\rangle$. Every policy π induces two value functions V_R^π and V_C^π. We will say two policies π_1 and π_2 are equivalent if $V_R^{\pi_1} = V_R^{\pi_2}$ and $V_C^{\pi_1} = V_C^{\pi_2}$ over all states $s \in S$. Let $\overline{\pi}$ denote the set of policies equivalent to π. Let $\overline{\pi}_1$ and $\overline{\pi}_2$ be two distinct equivalence classes of policies. We will say that $\overline{\pi}_1$ dominates $\overline{\pi}_2$ if $V_R^{\overline{\pi}_1}(s_0) \geq V_R^{\overline{\pi}_2}(s_0)$ and $V_C^{\overline{\pi}_1} \leq V_C^{\overline{\pi}_2}$. That is, $\overline{\pi}_1$ is superior in either R_R or R_C or both. An equivalence class is non-dominated if there does not exist an equivalence class that dominates it.

Let $\Pi(\tau)$ be the space of deterministic policies such that $\forall \pi \in \Pi(\tau), V_C^\pi(s_0) \leq \tau$. These are the feasible deterministic policies. An optimal feasible deterministic policy $\pi_\tau^* \in \Pi(\tau)$ satisfies

$$V_R^{\pi_\tau^*}(s_0) \geq V_R^\pi(s_0) \; \forall \pi \in \Pi(\tau).$$

Values are defined in the usual way as the expected cumulative discounted return:

$$V_C(s_0) = \mathbb{E}[\sum_t \gamma^t R_C(s_t, \pi(s_t))],$$

and

$$V_R(s_0) = \mathbb{E}[\sum \gamma^t R_R(s_t, \pi(s_t))].$$

An optimal feasible policy π_τ^* is not necessarily non-dominated. There might be another policy π' that achieves the same $V_R(s_0)$ but has larger $V_C^{\pi'}(s_0) > V_C^{\pi_\tau^*}(s_0)$ that is still feasible.

Define the Lagrangian MDP $\mathcal{L}(\lambda) = \langle S, A, P, \lambda R_R - (1-\lambda)R_C, \gamma, s_0\rangle$ whose reward function is a linear combination of R_R and R_C.

PAC-RL for Constrained MDPs

We now consider the problem of finding an approximately optimal policy by sampling from a simulator-defined Constrained MDP. We introduce the following parameters:

- τ defines the feasibility constraint. A policy π is feasible if $V_C^\pi(s_0) \leq \tau$.
- ϵ defines a tolerance on the optimality of $V_R^\pi(s_0)$.
- ν defines a tolerance on feasibility. We will accept any policy for which $|V_C^\pi(s_0) - V_C^*(s_0)| \leq \nu$, which means that in the worst case, $V_C^\pi(s_0) = \tau + \nu$.
- η controls the numerical precision of the λ values.
- δ is the confidence parameter.

Definition 4 *(Chang (2016)). A deterministic policy π is called ν-feasible if $V_C^\pi(s_0) \leq \tau + \nu$ for $\nu \geq 0$.*

Definition 5 *Let Π_L be the set of all stationary deterministic policies that are solutions to the Lagrangian MDP for some value of λ.*

Definition 6 *An algorithm is Lagrangian-PAC-SAFE-RL if, for any C-MDP $M(\tau) = \langle S, A, P, R_R, R_C, \tau, \gamma, s_0\rangle$ and any parameters $\epsilon > 0, \delta \in (0,1), \tau \in (0,1], \eta > 0$, and $\nu > 0$ the algorithm halts in time polynomial in $|S|, |A|, 1/(1-\gamma), 1/\epsilon, 1/\nu, 1/\delta$, and $1/\eta$ and does one of the following two things:*

1. *Outputs a policy $\pi \in \Pi_{\mathcal{L},\eta}$ such that with probability $1-\delta$ the following are simultaneously true:*
   (a) *$V_C^\pi(s_0) < \tau + \nu$ (π is $\tau + \nu$ feasible)*
   (b) *$V_R^{*(-\nu)}(s_0) - V_R^\pi(s_0) \leq \epsilon$ (the value of π is never less than ϵ below the value of the optimal $\tau - \nu$ feasible policy, and it may be significantly higher)*
2. *Outputs the message Fail, in which case with probability $1-\delta$ there does not exist any policy $\pi \in \Pi_{\mathcal{L},\eta}$ such that $V_C^\pi(s_0) \leq \tau + \nu$.*

This definition gives us control over how close to feasible the policy is (via ν) and how close to the optimal feasible policy its V_R return is (via ϵ).

Confidence intervals for V_R and V_C for policy evaluation

Suppose we have drawn a set of samples for various states and actions. For any fixed policy π, we can perform extended policy evaluation (i.e., extended value iteration with a fixed policy) to obtain lower and upper confidence bounds on $V_C(s_0)$ and $V_R(s_0)$. We will denote these as $\underline{V}_C^\pi(s_0)$, $\overline{V}_C^\pi(s_0)$, $\underline{V}_R^\pi(s_0)$, and $\overline{V}_R^\pi(s_0)$. Suppose our goal is to determine whether π is feasible and if it is, then to determine confidence intervals on $V_R^\pi(s_0)$. The policy π will be feasible with probability $1-\delta$ if $\overline{V}_C^\pi(s_0) \leq \tau$. Conversely, π is not feasible with probability $1-\delta$ if $\underline{V}_C^\pi(s_0) > \tau$.

Confidence intervals for V_R and V_C for policy optimization

Instead of using a fixed policy, we can set a value of λ and perform extended value iteration based on the upper confidence bound of the Lagrangian objective. This will define the $\pi^{UCB(\lambda)}$ policy. More generally, we can perform binary search on λ to find three values:

- $\underline{\lambda}$ is the largest value of $\lambda \in \Lambda$ such that $\overline{V}_C^{UCB(\lambda)}(s_0) \leq \tau$. This means that given our current sample, $\pi^{UCB(\lambda)}$ is the "best" policy (in the sense of having the largest λ) for which we can guarantee with probability $1 - \delta$ that it is feasible.
- $\overline{\lambda}$ is the largest value of $\lambda \in \Lambda$ such that $\underline{V}_C^{UCB(\lambda)}(s_0) \leq \tau$. This means that given our current sample, this is the largest value of λ that we cannot prove is not feasible.

The solid lines denote the true values of V_C and V_R. The dashed lines denote the corresponding upper and lower confidence bounds. For purposes of this section, let π^* be the policy in $\Pi(\tau, \eta)$ that maximizes $V_R^\pi(s_0)$. That is, π^* is τ-feasible and among all such policies it maximizes the V_R return.

Extended Value Iteration

Classical value iteration computes an optimal policy for a fixed MDP. Extended value iteration can compute optimal policy for finite-sampled optimistic/pessimistic MDPs by defining confidence intervals on the value function at each state of the MDP based on samples from that MDP. Different confidence interval methods (e.g., Hoeffding bound (Hoeffding, 1963), empirical Bernstein bound (Audibert, Munos, and Szepesvári, 2009), multinomial confidence interval (Weissman et al., 2003), etc.) at each state lead to different confidence intervals throughout the MDP. One can obtain robust policies from pessimistic MDPs (Tamar, Mannor, and Xu, 2014). Based on our experiments, the empirical Bernstein bound is the tightest bound compared to the other bounds.

The Empirical Bernstein Method: This approach uses the empirical Bernstein bound. Let $M(s, a)$ denote the sample mean of the discounted backed-up values from the successor states that result from taking action a in state s, and $Var(s, a)$ denote the sample variance of these values. We denote the upper and lower bounds on these values as $\overline{M}(s, a)$, $\underline{M}(s, a)$,$\overline{Var}(s)$, and $\underline{Var}(s)$.

$$\begin{aligned}\overline{M}(s,a) &= \sum_{s'} \hat{P}(s'|s,a)\gamma\overline{V}(s') \\ \overline{Var}(s,a) &= \sum_{s'} \hat{P}(s'|s,a)[\gamma\overline{V}(s') - \overline{M}(s,a)]^2 \\ \overline{V}(s) &= \max_a R(s,a) + \overline{M}(s,a) + \\ &\sqrt{\frac{2\overline{Var}(s)\ln(3/\delta_0)}{N(s,a)}} + \frac{3\gamma V_{max}\ln(3/\delta_0)}{N(s,a)}\end{aligned} \tag{4}$$

The lower bounds could be defined in a similar way as above. We need to define δ_0 so that the confidence intervals hold simultaneously with probability $1 - \delta$. These equations can be iterated to convergence. At convergence, with probability $1 - \delta$, $\underline{V}(s_0) \leq V^*(s_0) \leq \overline{V}(s_0)$.

Algorithm

The extended value iteration for the Lagrangian objective computes upper and lower bounds on V_R and V_C in all states and on $Q_R(s, a)$ and $Q_C(s, a)$ in all state-action pairs. A binary search algorithm (see supplementary materials) on λ finds $\underline{\lambda}$ and $\overline{\lambda}$ to within tolerance η for a given set of samples. We will apply BINARYSEARCH to find $\underline{\lambda}$ and $\overline{\lambda}$. For $\underline{\lambda}$, we are looking for the point λ where $\overline{V}_C^\lambda(s_0)$ crosses τ, which is exactly what BINARYSEARCH does. For $\overline{\lambda}$, we need to find the point where $\underline{V}_C^\lambda(s_0)$ crosses τ, determine the value on the larger side, and then find the largest value of $\overline{\lambda}$ that achieves that value. The function NEXTLARGERLAMBDA finds the next larger value of λ that will cause the UCB policy to change by calling LAGRANGIANEVI.

The main algorithm works by maintaining an upper bound $\overline{V}_R^{UCB(\overline{\lambda}^{(-\nu)})}(s_0)$ on the value of the best $(\tau - \nu)$-feasible policy and a lower bound $\underline{V}_R^{UCB(\underline{\lambda}^{(+\nu)})}(s_0)$ on the value of the best $(\tau + \nu)$-feasible policy. Here the notation $\lambda^{(-\nu)}$ refers the $(\tau - \nu)$ feasibility and $\lambda^{(+\nu)}$ refers to $(\tau + \nu)$ feasibility. Sampling proceeds in a series of minibatches that cause these bounds to shrink toward one another. Execution terminates when $\overline{V}_R^{UCB(\overline{\lambda}^{(-\nu)})}(s_0) - \underline{V}_R^{UCB(\underline{\lambda}^{(+\nu)})}(s_0) \leq \epsilon$. This is summarized in Algorithm1).

The rationale is the following. The largest value that $V_R^{*(-\nu)}(s_0)$ could have is $\overline{V}_R^{UCB(\overline{\lambda}^{(-\nu)})}(s_0)$. The smallest value that $\pi^{UCB(\underline{\lambda}^{(+\nu)})}$ could have is $\underline{V}_R^{UCB(\underline{\lambda}^{(+\nu)})}(s_0)$. We want the value of $\pi^{UCB\underline{\lambda}^{(+\nu)})}$ to be no less than ϵ below the value of $V_R^{*(-\nu)}(s_0)$. We attain this by ensuring that $\overline{V}_R^{UCB(\underline{\lambda}^{(+\nu)})}(s_0) - \underline{V}_R^{UCB(\overline{\lambda}^{(-\nu)})}(s_0) < \epsilon$.

Correctness and Polynomial Running Time

The proofs for the following claims and theorem are provided in supplementary materials.

Claim 1 *For any fixed λ, the optimal policy π^*_λ for $\mathcal{L}(\lambda)$ is a non-dominated policy.*

Claim 2 *Let λ_1 and λ_2 be a pair of values such that $\lambda_2 = \lambda_1 - \delta$ for some positive δ. Let π_1 be a policy that optimizes the Lagrangian for $\lambda = \lambda_1$ and π_2 be the policy that optimizes the Lagrangian for $\lambda = \lambda_2$. Then one of two cases holds:*

Case 1: *$V_C^{\pi_2}(s_0) = V_C^{\pi_1}(s_0)$, and $V_R^{\pi_2}(s_0) = V_R^{\pi_1}(s_0)$ or*

Case 2: *$\pi_1 \neq \pi_2$, $V_C^{\pi_2}(s_0) < V_C^{\pi_1}(s_0)$, and $V_R^{\pi_2}(s_0) < V_R^{\pi_1}(s_0)$.*

Claim 3 *There exists a value λ^* such that $\forall \lambda \leq \lambda^*$, the optimal policy, π^*_λ, of the Lagrangian MDP $\mathcal{L}(\lambda)$ is feasible for $\mathcal{M}(\tau)$; that is $V_C^{\pi^*_\lambda}(s_0) \leq \tau$.*

For computational efficiency, we will not consider all possible values of λ. Instead, we discretize the space by introducing a precision parameter η. Define $\Pi_{\mathcal{L},\eta}$ to be the class of all policies in $\Pi_{\mathcal{L}}$ where $\lambda = k\eta$, for $k \in \{0, 1, \ldots, 1/\eta\}$. We will restrict our attention to only these policies.

To obtain a polynomial time sampling algorithm, we need to relax our goal (based on ideas from Chang (2016)). Let $\Pi_{\mathcal{L},\eta}(\tau)$ be the set of all policies $\pi \in \Pi_{\mathcal{L},\eta}$ such that $V_C^\pi(s_0) \leq \tau$. These are the τ-feasible policies. We will

Algorithm 1: CONSTRAINEDDDV$(s_0, \tau, \nu, F, \epsilon, \delta, \gamma, R_{max})$

```
1: $\underline{\lambda}^{(+\nu)} := 0; \overline{\lambda}^{(-\nu)} := 1$
2: CheckFeasibility:=true
3: **loop**
4:    $\overline{\lambda}^{(-\nu)}$ = FINDUPPER$(0, 1, \max(0, \tau - \nu), \eta)$
5:    $\underline{\lambda}^{(+\nu)}$ = FINDLOWER$(0, 1, \min(1, \tau + \nu), \eta)$
6:    **if** CheckFeasibility **then**
7:       LAGRANGIANEVI$(0, \eta, \delta)$
8:       **if** $\underline{V}_C^{UCB(0)}(s_0) \geq \tau - \nu$ **then**
9:          {there is no $(\tau - \nu)$-feasible policy}
10:         **return** No feasible policy
11:      **else if** $\overline{V}_C^{UCB(0)}(s_0) < \tau - \nu$ **then**
12:         {there is a $(\tau - \nu)$-feasible policy}
13:         CheckFeasibility:=false
14:      **end if**
15:   **end if**
16:   **if** $\left(\overline{\lambda}^{(-\nu)} = \underline{\lambda}^{(+\nu)}\right)$ **and**
      $\left(\overline{V}_R^{UCB(\overline{\lambda}^{(-\nu)})}(s_0) - \underline{V}_R^{UCB(\underline{\lambda}^{(+\nu)})}(s_0) \leq \epsilon\right)$ **then**
17:      **return** $\left(Success, \pi^{UCB(\underline{\lambda}^{(+\nu)})}\right)$
18:   **end if**
19:   Explore for a minibatch of $B$ samples using DDV
      on $\pi^{UCB(\overline{\lambda}^{(-\nu)})}$
20: **end loop**
```

be interested in two other policy classes: $\Pi_{\mathcal{L},\eta}(\tau - \nu)$ and $\Pi_{\mathcal{L},\eta}(\tau + \nu)$.

Let $\pi^{*(-\nu)} \in \Pi_{\mathcal{L},\eta}(\tau - \nu)$ be a policy that is feasible with respect to the threshold $\tau - \nu$ and that among all such policies maximizes $V_R(s_0)$. More precisely, $\pi^{*(-\nu)} = \arg\max_{\pi \in \Pi_{\mathcal{L},\eta}(\tau-\nu)} V_R^{\pi}(s_0)$.

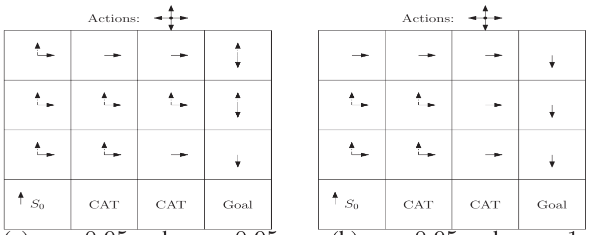
Denote the value of $\pi^{*(-\nu)}$ by $V_R^{*(-\nu)}(s_0)$. Our goal will be to output a policy $\pi \in \Pi_{\mathcal{L},\eta}(\tau + \nu)$ such that $V_R^{*(-\nu)}(s_0) - V_R^{\pi}(s_0) \leq \epsilon$ and to do so in polynomial time.

Claim 4 *The optimal value* $\lambda^* \in [\underline{\lambda}, \overline{\lambda}]$ *with probability* $1 - \delta$.

Claim 5 $\underline{V}_R^{UCB(\underline{\lambda})}(s_0) \leq V_R^*(s_0) \leq \overline{V}_R^{UCB(\overline{\lambda})}(s_0)$ *with probability* $1 - \delta$.

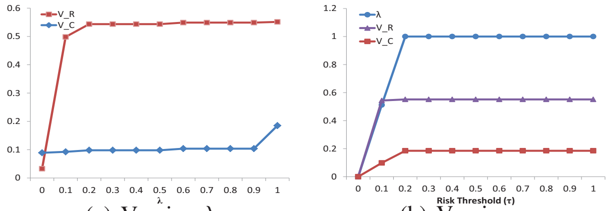
Note that the gap between $\overline{V}_R^{UCB(\overline{\lambda})}(s_0)$ and $\underline{V}_R^{UCB(\underline{\lambda})}(s_0)$ is composed of three parts. First, there is the width of the upper confidence interval $\overline{V}_R^{UCB(\overline{\lambda})}(s_0) - V_R^{UCB(\overline{\lambda})}(s_0)$. Second, there is the difference in the values of the policies $\pi^{UCB(\overline{\lambda})}$ and $\pi^{UCB(\underline{\lambda})}$, which we can write as $V_R^{UCB(\overline{\lambda})}(s_0) - V_R^{UCB(\underline{\lambda})}(s_0)$. Finally, there is the width of the lower confidence interval $V_R^{UCB(\underline{\lambda})}(s_0) - \underline{V}_R^{UCB(\underline{\lambda})}(s_0)$.

To prove correctness, we must show that, under appropriate conditions, the CONSTRAINEDDDV algorithm will terminate at line 17. Specifically, we will prove the following claim:



(a) $\gamma = 0.95$ and $\gamma_C = 0.95$ (b) $\gamma = 0.95$ and $\gamma_C = 1$

Figure 1: Derived policies for the GridWorld domain; solid arrows are when $\lambda = 1$ and dotted arrows are when $\lambda = 0$. When both policies agree on an action in a cell, only one is shown.



(a) Varying λ (b) Varying τ

Figure 2: Value of reward and risk while varying λ and risk threshold (τ) for the GridWorld domain.

Claim 6 *If* $\Pi_{\mathcal{L},\eta}(\tau - \nu)$ *and* $\Pi_{\mathcal{L},\eta}(\tau + \nu)$ *are non-empty and* $0 < \lambda^* < 1$, *then with probability* $1 - \delta$, CONSTRAINEDDDV *will terminate at line 17.*

We can also show the following.

Claim 7 *If there is no* $(\tau - \nu)$*-feasible policy, then the* CONSTRAINEDDDV *algorithm will terminate at line 10.*

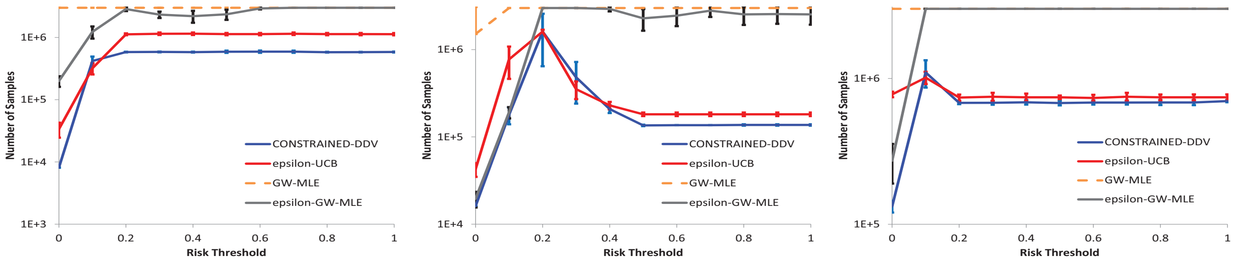
Theorem 1 CONSTRAINEDDDV *requires polynomial sample size and terminates in polynomial computation time.*

Experiments

We report three experiments. First, we study the GridWord domain shown in Figure 1(a) (there is one starting state, one goal state, and two catastrophic states). Our goal is to gain some intuition about the C-MDP formulation. Specifically, we look at the policies for $\lambda = 0$ and $\lambda = 1$.

In Figure 1, we assume the model is known. The solid lines show the optimal policy for $\lambda = 1$ (maximizing the reward), and the dotted actions show the optimal policy for $\lambda = 0$ (minimizing the risk). Notice that even for unequal discount factors, we are able to find a desirable policy, which may not be optimal. The main difference between the policies for discounted and undiscounted risk is that for discounted risk the best stationary deterministic policy that minimizes the risk takes the discount into account and moves toward the goal more slowly than the undiscounted risk policy.

In the second experiment, we solve for the optimal policy when the MDP is known while varying λ and the constraint



(a) GridWorld (b) Tamarisk $R = 3$ and $H = 1$ (c) Tamarisk $R = 3$ and $H = 2$

Figure 3: Comparison of number of samples taken by each algorithm to reach to the termination point.

threshold τ. Our goal is to determine the right answer and see the impact of τ and λ. Figure 2 shows the value of reward (V_R) and value of risk (V_C) in the starting state for the GridWorld domain while varying the value of λ (2(a)) and while varying the value of τ (2(b)). There is no feasible policy when $\tau = 0$.

In Figure 2(a), we see that when λ is close to 1, we can easily reduce V_C without any impact on V_R. As λ shrinks, V_C and V_R both shrink gradually, so that for values of τ in the range (0.185 to 0.1), there continues to be little impact on V_R. However, when λ goes from 0.1 to 0.0, we see a huge drop in V_R for very little gain in V_C. This kind of sudden drop causes difficulty for obtaining PAC results. The problem is that in this region, the confidence intervals on V_R will be very wide, and it can require a huge number of training samples to shrink them enough to achieve a width of ϵ.

In the third experiment, we compare the sample complexity of CONSTRAINEDDDV against three benchmark algorithms: GW-MLE, ϵ_g-greedy GW-MLE, ϵ_g-greedy UCB. GW-MLE is the improved version of the algorithm of Geibel and Wysotzki (2005), which basically maximizes the Lagrangian defined as $\mathcal{L}(\hat{\lambda}) = \langle S, A, \hat{P}, \hat{\lambda}R_R - (1 - \hat{\lambda})R_C, \gamma, s_0\rangle$, where $\hat{\lambda}$ is the maximum likelihood estimate of λ calculated over the MDP with transition probability $\hat{P}$ and reward functions R_R and R_C. The GW-MLE algorithm samples along the induced $\pi^{\hat{\lambda}}$ policy at each mini-batch. UCB algorithm calculates $\pi^{UCB} = \arg\max_a \overline{Q}_R(s, a)$ and samples along the π^{UCB} policy. Since the UCB algorithm ignores the risk in its default operation, we have added an adjustable ϵ_g parameter for better exploration. The algorithms are modified to have stopping condition similar to the lines 8 and 16 in Algorithm 1 .

We compared these algorithms on the GridWorld MDP and two instances of the tamarisk domain. In these experiments, we learn the model by sampling from the simulator. Tamarisk problem instances are configured with the number of river segments ($E = 3$) and the number of slots ($H = 1$) and ($H = 2$) (for more detail see Taleghan et al. (2015)). For the ($E = 3, H = 1$) problem, the starting state was NTE (one site contains a native species, one is invaded by tamarisk, and one site at the bottom of river is empty). For the ($E = 3, H = 2$) instance, the starting state is NTEEEE (one site contains a native species and an invasive species and the rest of the sites in the river are empty). A catastrophic state is any state in which there are no natives (species extinction). The goal state is that all sites are fully occupied by native species. We optimized the value of ϵ_g for ϵ_g-greedy GW-MLE and ϵ_g-greedy UCB algorithms among the candidate values $\epsilon_g \in \{0.01, 0.1, 0.25\}$. After sampling a minibatch of size $B = 1000$ we update the model and calculate the corresponding confidence bounds. We calculate $\overline{\lambda}$ and $\underline{\lambda}$ every 8000 samples.

In these experiments, $\gamma = \gamma_C = 0.95$, $\delta = 0.01$, $\eta = 0.01$, and $\nu = 0.025$. For the GridWorld domain, $\epsilon = 0.2$, and for the Tamarisk problems $\epsilon = 1$. The algorithms terminate either if the width of the confidence interval falls below ϵR_{max} or if 3 million samples are drawn.

We report the number of samples drawn at termination in Figure 3. The results are averaged over 10 independent runs, and the vertical axis is plotted on a log scale. Error bars indicate one standard deviation. The GW-MLE and ϵ_g-GW-MLE algorithms perform very poorly; much worse than CONSTRAINEDDDV. In many cases, they hit the 3 million maximum sampling budget without achieving the desired confidence interval width. CONSTRAINEDDDV and ϵ_g-UCB give much more similar performance, if ϵ_g is properly tuned. CONSTRAINEDDDV almost always requires smaller sample sizes, particularly for small values of τ (which would be the values normally encountered in a real application).

Conclusion

Many computational sustainability problems involving MDPs must be concerned with catastrophic outcomes such as species extinction. One approach to this is to limit the probability of catastrophic outcomes by imposing a constraint on the MDP policy, which converts the MDP into a Constrained MDP (C-MDP). Previous work on simulation-based MDP planning for constrained MDPs has not provided formal guarantees. This paper is the first to provide an algorithm with formal guarantees by extending the notion of PAC-RL algorithms to PAC-Safe-RL algorithms. We proved that this new algorithm, CONSTRAINEDDDV, is PAC-Safe-RL. Our experiments demonstrated that CONSTRAINEDDDV is also able to match or beat the sample complexity of very competitive baseline algorithms that lack formal performance guarantees.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant Nos. 1331932 and 1521687.

References

Altman, E. 1999. *Constrained Markov decision processes*, volume 7. CRC Press.

Audibert, J. Y.; Munos, R.; and Szepesvári, C. 2009. Exploration-exploitation trade-off using variance estimates in multi-armed bandits. *Theoretical Computer Science* 410(19):1876–1902.

Boutilier, C., and Lu, T. 2016. Budget allocation using weakly coupled, constrained markov decision processes. https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/45291.pdf.

Chang, H. S. 2016. Sleeping experts and bandits approach to constrained markov decision processes. *Automatica* 63:182 – 186.

Dietterich, T. G.; Alkaee Taleghan, M.; and Crowley, M. 2013. PAC optimal planning for invasive species management: improved exploration for reinforcement learning from simulator-defined MDPs. In *Association for the Advancement of Artificial Intelligence AAAI 2013 Conference (AAAI-2013)*.

Dolgov, D. A., and Durfee, E. H. 2005. Stationary deterministic policies for constrained mdps with multiple rewards, costs, and discount factors. In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI-05)*, 1326–1332.

Feinberg, E. A., and Rothblum, U. G. 2012. Splitting randomized stationary policies in total-reward markov decision processes. *Mathematics of Operations Research* 37(1):129–153.

Feinberg, E. A., and Shwartz, A. 1996. Constrained discounted dynamic programming. *Mathematics of Operations Research* 21(4):922–945.

Feinberg, E. A. 2000. Constrained discounted markov decision processes and hamiltonian cycles. *Mathematics of Operations Research* 25(1):130–140.

Feinberg, E. A. 2008. Optimality of deterministic policies for certain stochastic control problems with multiple criteria and constraints. In *Mathematical Control Theory and Finance*. Springer. 137–148.

Fiechter, C.-N. 1994. Efficient reinforcement learning. In *Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory*, 88–97. ACM Press.

García, J., and Fernández, F. 2015. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research* 16:1437–1480.

Geibel, P., and Wysotzki, F. 2005. Risk-sensitive reinforcement learning applied to control under constraints. *J. Artif. Intell. Res.(JAIR)* 24:81–108.

Geramifard, A. 2012. *Practical Reinforcement Learning Using Representation Learning And Safe Exploration For Large Scale Markov Decision Processes*. Ph.D. Dissertation, Massachusetts Institute of Technology.

Hoeffding, W. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* 58(301):13–30.

Sheldon, D.; Dilkina, B.; Elmachtoub, A.; Finseth, R.; Sabharwal, A.; Conrad, J.; Gomes, C.; Shmoys, D.; Allen, W.; and Amundsen, O. 2010. Maximizing the spread of cascades using network design. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 517–526.

Strehl, A., and Littman, M. 2008. An analysis of model-based interval estimation for Markov decision processes. *Journal of Computer and System Sciences* 74(8):1309–1331.

Syed, U.; Bowling, M.; and Schapire, R. 2008. Apprenticeship learning using linear programming. In *International Conference on Machine Learning*.

Taleghan, M. A.; Dietterich, T. G.; Crowley, M.; Hall, K.; and Albers, H. J. 2015. PAC optimal MDP planning with application to invasive species management. *Journal of Machine Learning Research* 16:3877–3903.

Tamar, A.; Mannor, S.; and Xu, H. 2014. Scaling up robust MDPs using function approximation. In *ICML 2014*, volume 32.

Weissman, T.; Ordentlich, E.; Seroussi, G.; Verdu, S.; and Weinberger, M. J. 2003. Inequalities for the L1 deviation of the empirical distribution. Technical report, HP Labs.

Zadorojniy, A.; Even, G.; and Shwartz, A. 2009. A strongly polynomial algorithm for controlled queues. *Mathematics of Operations Research* 34(4):992–1007.

Towards a Data Efficient Off-Policy Policy Gradient

Josiah P. Hanna, Peter Stone
Dept. of Computer Science
The University of Texas at Austin
Austin, TX 78712 USA
{jphanna,pstone}@cs.utexas.edu

Abstract

The ability to learn from off-policy data – data generated from past interaction with the environment – is essential to data efficient reinforcement learning. Recent work has shown that the use of off-policy data not only allows the re-use of data but can even improve performance in comparison to on-policy reinforcement learning. In this work we investigate if a recently proposed method for learning a better data generation policy, commonly called a behavior policy, can also increase the data efficiency of policy gradient reinforcement learning. Empirical results demonstrate that with an appropriately selected behavior policy we can estimate the policy gradient more accurately. The results also motivate further work into developing methods for adapting the behavior policy as the policy we are learning changes.

Introduction

Off-policy RL is necessary for data efficient reinforcement learning. The standard way to incorporate off-policy data into reinforcement learning is to use importance sampling. Unfortunately, policy improvement with importance sampling may exhibit instability due to increased variance (Levine and Koltun 2013; Thomas, Theocharous, and Ghavamzadeh 2015). Recent work has shown that importance sampling can actually lead to more data efficient policy evaluation (Hanna et al. 2017). This work introduced a method called behavior policy gradient (BPG) and demonstrated it can find data generation policies that give low variance importance sampling evaluations. Here we investigate the problem of policy improvement with a data generation policy that has been learned with BPG. Specifically, we investigate whether a behavior policy that gives low variance evaluation of an initial policy can also be used to effectively estimate the direction of the policy gradient and if this same policy can be used for multiple policy gradient updates. Empirical results show that 1) off-policy policy gradient estimates with such a behavior policy lead to larger performance gains with a single update and 2) that this improvement can be realized for a limited number of policy improvement steps before off-policy gradient estimates lead to worse performance than on-policy gradient estimates.

Preliminaries

We assume the environment is represented as a finite horizon, episodic MDP. The agent interacts with the environment in a series of episodes by selecting actions from a policy π. Each episode can be described as a trajectory, H, that consists of a sequence of states, actions, and rewards: $H = S_0, A_0, R_0, ...S_L, A_L, R_L$. The return of a trajectory, denoted $g(h)$, is the sum of rewards along the trajectory: $g(H) = \sum_{t=0}^{L} R_t$. We assume π is a parameterized, stochastic policy with parameter vector $\boldsymbol{\theta}$ and write $H \sim \pi$ to denote sampling a trajectory by following policy π for one episode. The expected return of a policy, π, is $J(\pi) = \mathbf{E}[g(H)|H \sim \pi]$.

In reinforcement learning, policy improvement is the iterative process of updating a policy towards a policy with higher expected return. Denote the initial policy as $\pi_{\boldsymbol{\theta}_0}$. At step i a policy improvement method updates $\boldsymbol{\theta}_i$ to $\boldsymbol{\theta}_{i+1}$ such that $J(\pi_{\boldsymbol{\theta}_{i+1}}) > J(\pi_{\boldsymbol{\theta}_i})$. Policy improvement can continue for a fixed number of iterations or until there is no longer an increase in the expected return.

Naturally, policy improvement requires interaction with the environment. We will refer to the policy that generates the trajectories for a step of policy improvement as the *behavior policy*. The policy being updated is the *target policy*. Methods where the target policy is also the behavior policy are termed *on-policy*; otherwise, they are *off-policy*.

Policy Gradient Reinforcement Learning Policy gradient methods are a popular class of reinforcement learning algorithms used for policy improvement (Deisenroth et al. 2013). Policy gradient methods attempt to maximize the expected return of a policy $\pi_{\boldsymbol{\theta}}$ with respect to the policy parameters $\boldsymbol{\theta}$. This gradient can be derived as:

$$\frac{\partial}{\partial \boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) = \mathbf{E}\left[g(H) \sum_{t=0}^{L} \frac{\partial}{\partial \boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A_t|S_t) \right] \tag{1}$$

where $H \sim \pi_{\boldsymbol{\theta}}$. The simplest policy gradient method is the REINFORCE algorithm which adapts the policy with unbiased estimates of Eq. 1 (Williams 1992). In this form, estimates of the policy gradient often suffer from high variance. Extensive work has gone in to reducing this variance in order to scale policy gradient methods to complex problems (e.g., (Peters, Mülling, and Altun 2010; Greensmith et al. 2001; Schulman et al. 2015; 2016; Gu et al. 2017)). As a result, policy gradient methods are a widely applied class of RL algorithms.

Note that policy gradient methods are typically on-policy methods in that we estimate the gradient at π with trajectories sampled from π. In practice this means that at step i of learning, policy π_i is used to collect a dataset of trajectories, $\mathcal{D}_i$, $\mathcal{D}_i$ is used to estimate (1), a gradient step is taken on $\boldsymbol{\theta}_i$, and then $\mathcal{D}_i$ *is discarded* and the process repeats with policy π_{i+1}.

Behavior Policy Search This section describes a recently proposed off-policy method for policy evaluation that uses *importance sampling* to lower the variance of policy evaluation. In the next section we will adapt this idea to the policy gradient setting.

Consider the policy evaluation setting where our goal is to evaluate a *target policy*, π. The simplest approach is to execute π for multiple episodes and average the resulting returns. Unfortunately, this Monte Carlo estimator may have high variance when the target policy rarely experiences trajectories with high-magnitude return.

Instead of running π, we can instead run a different behavior policy, π_b and weight the resulting returns according to the likelihood of seeing them under π instead of π_b. This approach allows us to over-sample these rare, high-magnitude returns and then weight them according to their true likelihood. Importance sampling is an *unbiased* method for computing the re-weighting. The importance sampled return of a trajectory H is:

$$\mathrm{IS}(H, \pi_b) = \prod_{t=0}^{L} \frac{\pi(A_t|S_t}{\pi_b(A_t|S_t)} \cdot g(H)$$

Given a dataset of trajectories, $\mathcal{D}$, generated by π_b the importance sampling estimator is the mean of $\mathrm{IS}(H, \pi_b)$ over all $H \in \mathcal{D}$.

Recent work by Hanna et al. demonstrated that it is possible to find a behavior policy that leads to lower variance policy evaluation compared to Monte Carlo policy evaluation (Hanna et al. 2017). Their *behavior policy gradient* (BPG) method used gradient descent on the variance of the importance sampling estimator to adapt a parameterized behavior policy towards a locally optimal behavior policy.

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i + \alpha \mathbf{E}\left[\mathrm{IS}(H, \pi_{\boldsymbol{\theta}})^2 \sum_{t=0}^{L-1} \frac{\partial}{\partial \boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A_t|S_t)\right]$$

where $H \sim \pi_{\boldsymbol{\theta}}$. The result of running BPG for a particular target policy π is a behavior policy, π_b, that generates data for low variance importance sampling evaluation of a π. This low variance evaluation is only guaranteed for a static target policy.

Off-Policy Policy Gradient

This section discusses how we can apply behavior policy search to policy gradient methods. While there have been many important contributions since Williams' original REINFORCE work, we will primarily discuss REINFORCE and note that other approaches (e.g., optimal baselines (Greensmith et al. 2001; Peters and Schaal 2008), trust-regions (Peters, Mülling, and Altun 2010; Schulman et al. 2015), etc.) could be combined with the presented approach in future work.

The REINFORCE method can be adapted to an off-policy variant by using unbiased estimates of an importance-sampled version of Equation 1

$$\frac{\partial}{\partial \boldsymbol{\theta}} J(\pi_{\boldsymbol{\theta}}) = \mathbf{E}\left[\mathrm{IS}(H, \pi_b) \sum_{t=0}^{L} \frac{\partial}{\partial \boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(A_t|S_t)\right] \qquad (2)$$

where $H \sim \pi_b$. As in policy evaluation, if π_b is chosen arbitrarily gradient estimates are likely to have high variance. On the other hand, if we can select π_b appropriately then our gradient estimate may have less variance than the on-policy version.

We will select π_b to be a behavior policy that minimizes the variance of an importance sampling evaluation of the current policy. This approach allows us to directly apply BPG to learn π_b. In contrast, previous work has considered the trace of the gradient covariance matrix as the measure of gradient variance (Peters and Schaal 2008; Gu et al. 2017; Ciosek and Whiteson 2017; Bouchard et al. 2016). Minimizing this variance measure is equivalent to minimizing the variance of each component of the gradient. This measure has been used in previous work on adaptive importance sampling for stochastic gradient descent (Bouchard et al. 2016; Ciosek and Whiteson 2017). One downside of this measure is that it may be sensitive to the scale of the policy parameterization. Minimizing the variance of policy evaluation is scale-invariant although it is *not* guaranteed to lower policy gradient variance.

Another challenge for developing an off-policy REINFORCE method is the need to track the current policy. If we start with π_b that gives low variance policy gradient estimates for the initial policy it may not give low variance estimates after the initial policy has changed. One of our experiments attempts to evaluate the scale of this problem.

Empirical Results

We present two experiments using the Cartpole domain implented in OpenAI gym (Brockman et al. 2016). The policy is a softmax distribution over actions where the logits come from a linear combination of state variables. The initial behavior policy is trained with BPG to minimize the variance of an importance sampling evaluation of the initial policy. We design experiments to answer the questions 1) does a behavior policy selected with BPG lead to better estimation of the policy gradient direction and 2) can a behavior policy selected with BPG be used for multiple policy gradient updates?

Policy Improvement Step Quality

Our first experiment compares the quality of the update direction computed with an off-policy REINFORCE method to the quality of the update direction computed with standard REINFORCE. In order to make this comparison, we sample a batch of trajectories with the initial policy and another batch with π_b. We estimate the on-policy REINFORCE gradient, the off-policy REINFORCE gradient estimated with a behavior policy trained with BPG to evaluate the initial policy,

Method	Average Return (std.)
Random π_b	54.92 (8.27)
On-policy	55.081 (1.31)
Optimized π_b	**68.656 (15.7)**

Table 1: Comparison of one-step improvement in average return when estimating the policy gradient with off-policy and on-policy policy REINFORCE. For each behavior policy we sample 200 trajectories and estimate the policy gradient direction with (2). The gradient step size is computed with a line search. Results are averaged over 50 independent runs.

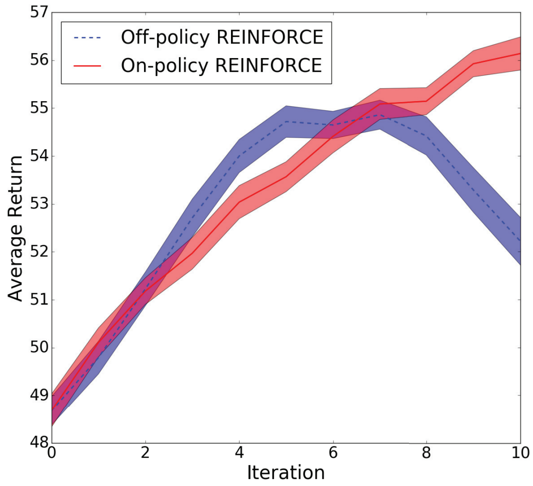


Figure 1: Comparison of multi-step improvement in average return when estimating the policy gradient with off-policy and on-policy REINFORCE.

and the off-policy REINFORCE gradient estimated with a randomly initialized behavior policy. For each method we select the optimal step-size for each method with a line search on $v(\pi)$. We use a line search to avoid conflating gradient direction with gradient magnitude.

Table 1 shows that the average gradient direction computed with off-policy REINFORCE leads to a much larger increase in expected return. However, we also point out that the variance of the performance improvement is also higher. While in most cases expected performance increases above the increase obtained by the other methods, the fact that the variance of the improvement has increased may suggest that lowering the variance of policy evaluation does *not* necessarily lead to a lower variance policy gradient estimate.

Multi-step Policy Improvement

Our second experiment investigates if a behavior policy trained to evaluate the initial policy can be used to estimate the policy gradient at other policies. For this experiment, we collect a single set of 100 trajectories with the behavior policy and adapt the target policy with off-policy REINFORCE for 10 iterations.

Figure 1 demonstrate that an improved π_b for importance sampling evaluation can lead to faster learning compared to on-policy REINFORCE – even without re-sampling new trajectories. However, the improvement is relegated to the first few iterations of policy improvement before the target policy has changed significantly.

Discussion and Open Questions

Our empirical results have shown that off-policy policy gradient estimates can give a more accurate estimate of the direction of the policy gradient better than on-policy policy gradient estimates. Our results also show that off-policy REINFORCE with a behavior policy trained with BPG can lead to faster initial learning but that performance degrades once the current policy has been adapted away from the initial policy. In order to develop a complete, low variance off-policy REINFORCE method it will be important to address the question of how to adapt the behavior policy so that it continues to lower variance as the current policy changes.

An alternative to adapting the behavior policy to track the current policy is to start with a behavior policy that generalizes to other policies along the trajectory of learning. One approach towards finding such a policy would be to regularize BPG so that it does not overfit to the policy it is trained to evaluate or to use meta-learning techniques to learn a behavior policy that can be quickly adapted to estimate the policy gradient for a new target policy (Finn, Abbeel, and Levine 2017).

Conclusion

We have presented preliminary steps towards a policy gradient algorithm that uses off-policy data for more efficient updates. We have described how a recently proposed behavior policy search method could be adapted to the policy improvement setting. We then presented experiments showing that a carefully selected behavior policy can improve the step direction of the REINFORCE method and that this same behavior policy can be used for multiple updates before it performs worse than an on-policy update. These results indicate that research into how to adapt the behavior policy as the policy being learned changes has the potential to further improve the data efficiency of policy gradient reinforcement learning.

Acknowledgements

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-1637736, IIS-1651089, IIS-1724157), Intel, Raytheon, and Lockheed Martin. Peter Stone serves on the Board of Directors of Cogitai, Inc. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

References

Bouchard, G.; Trouillon, T.; Perez, J.; and Gaidon, A. 2016. Online learning to sample. *arXiv preprint arXiv:1506.09016*.

Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; and Zaremba, W. 2016. Openai gym. *arXiv preprint arXiv:1606.01540*.

Ciosek, K., and Whiteson, S. 2017. OFFER: Off-environment reinforcement learning. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*.

Deisenroth, M. P.; Neumann, G.; Peters, J.; et al. 2013. A survey on policy search for robotics. *Foundations and Trends® in Robotics* 2(1–2):1–142.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

Greensmith, E.; Bartlett, P. L.; Baxter, J.; et al. 2001. Variance reduction techniques for gradient estimates in reinforcement learning. In *Proceedings of the 14th Conference on Neural Information Processing Systems (NIPS)*, 1507–1514.

Gu, S.; Lillicrap, T.; Ghahramani, Z.; Turner, R. E.; and Levine, S. 2017. Q-prop: Sample-efficient policy gradient with an off-policy critic.

Hanna, J.; Thomas, P. S.; Stone, P.; and Niekum, S. 2017. Data-efficient policy evaluation through behavior policy search. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

Levine, S., and Koltun, V. 2013. Guided policy search. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 1–9.

Peters, J., and Schaal, S. 2008. Reinforcement learning of motor skills with policy gradients. *Neural networks* 21(4):682–697.

Peters, J.; Mülling, K.; and Altun, Y. 2010. Relative entropy policy search. In *AAAI*, 1607–1612. Atlanta.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 1889–1897.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2016. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*.

Thomas, P. S.; Theocharous, G.; and Ghavamzadeh, M. 2015. High confidence policy improvement. In *Proceedings of the 32nd International Conference on Machine Learning, ICML*.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Bayesian Q-learning with Assumed Density Filtering

Heejin Jeong, Daniel D. Lee
Department of Electrical and Systems Engineering
University of Pennsylvania
Philadelphia, PA 19104
{heejinj, ddlee}@seas.upenn.edu

Abstract

While off-policy temporal difference methods have been broadly used in reinforcement learning due to their efficiency and simple implementation, their Bayesian counterparts have been relatively understudied. This is mainly because the max operator in the Bellman optimality equation brings non-linearity and inconsistent distributions over value function. In this paper, we introduce a new Bayesian approach to off-policy TD methods using *Assumed Density Filtering*, called *ADFQ*, which updates beliefs on action-values (Q) through an online Bayesian inference method. Uncertainty measures in the beliefs not only are used in exploration but they provide a natural regularization in the belief updates. We also present a connection between ADFQ and Q-learning. Our empirical results show the proposed ADFQ algorithms outperform comparing algorithms in several task domains. Moreover, our algorithms improve general drawbacks in BRL such as efficiency, usage of uncertainty, and nonlinearity.

Introduction

In reinforcement learning (RL), a learning subject seeks an optimal behavior by interacting with a dynamic environment which maximizes the *value* of each state: a sum of expected future outcomes starting from the state. Bayesian Reinforcement Learning (BRL) is one of the approaches in RL that deploys Bayesian inference in order to incorporate new information into prior information. It explicitly quantifies the uncertainty of learning parameters unlike standard RL algorithms which use point estimates of the parameters. Therefore, an explicit quantification of the uncertainty can optimize the exploration-exploitation trade-off by exploring actions with higher uncertainty more often than actions with lower uncertainty. Moreover, it can naturally regulate the posterior updates from the new information.

Utilizing such advantages, various algorithms have been proposed in both model-based BRL (Dearden, Friedman, and Andre 1999; Strens 2000; Poupart et al. 2006; Duff 2002; Guez, Silver, and Dayan 2012) and model-free BRL (Dearden, Friedman, and Russell 1998; Engel, Mannor, and Meir 2003; 2005; Geist and Olivier 2010; Chowdhary et al. 2014; Ghavamzadeh and Engel 2006). However, to our knowledge, only few Bayesian approaches to *off-policy temporal difference (TD) learning* have been studied compared to other methods due to the non-linearity in the Bellman optimality equation. Yet off-policy TD methods have been widely used in the standard RL. One of the most recent influential algorithms in Bayesian off-policy TD learning would be KTD-Q extended from Kalman Temporal Difference (KTD) (Geist and Olivier 2010). KTD approximates the value function using the Kalman filtering scheme. It considers parameters of the value function as hidden states and tracks them through indirect observations, or rewards from the environment. The KTD framework is applied to an off-policy TD algorithm (KTD-Q) as well as other TD algorithms (KTD-V and KTD-SARSA). They solved the non-linearity in the Bellman optimality equation by applying the Unscented Transform. Although the KTD framework handles some important features in RL, it requires many parameter values to be chosen and it is computationally expensive. Another limitation of KTD-Q is that it was proposed under a deterministic environment assumption and it was not extended for a stochastic environment case.

This paper presents a novel approximated Bayesian off-policy TD learning algorithm, termed as ADFQ, in finite state and action spaces which updates beliefs on Q-values and approximates their posteriors using an online Bayesian inference algorithm known as assumed density filtering (ADF). ADF, also known as *moment matching*, *online Bayesian learning*, and *weak marginalization*, has been proposed independently in several fields (Opper 1999; Boyen and Koller 1998; Maybeck 1982). It is a general technique for approximating a true posterior to a tractable parametric distribution in Bayesian networks. In the proposed ADFQ algorithms, ADF is used to solve the problem of the posterior inconsistency caused by the max operator in the Bellman optimality equation. We proposed two variants in ADFQ, ADFQ-Numeric and ADFQ-Approx, in terms of a way of computing approximation statistics. We experimented our algorithms on four different discrete domains, and compared them with Q-learning and KTD-Q. It consistently outperformed the comparing algorithms on all the domains. We showed that ADFQ improved some of major drawbacks of BRL such as computational complexity as well as that Q-learning could be a special case of ADFQ-Approx.

Background

Assumed Density Filtering

Suppose that a hidden variable $\mathbf{x}$ follows a tractable parametric distribution $p(\mathbf{x}|\theta_t)$ where θ_t is a set of parameters at time t. In the Bayesian framework, the distribution can be updated after new observation data (D_t) is drawn using the Bayes rule:

$$\hat{p}(\mathbf{x}|\theta_t, D_t) \propto p(D_t|\mathbf{x}, \theta_t)p(\mathbf{x}|\theta_t)$$

In online learning, this update happens every time a new data point is observed and the updated posterior is used as a prior in the following step and so on. The previous data is discarded after used.

When the updated posterior does not belong to its original parametric family, it has to be approximated to a distribution belonging to the family in order to continue the online learning. In ADF, the closest distribution in the family to the posterior is chosen by minimizing the reverse *Kullback-Leibler* divergence, a measure of the dissimilarity between the distributions denoted as $KL(\hat{p}||p)$ where $\hat{p}$ is an intractable distribution and p is a distribution in a parametric family of interest. Thus, in our example, this is applied as:

$$\theta_{t+1} = \underset{\theta}{\operatorname{argmin}}\, KL(\hat{p}(\cdot|\theta_t, D_t)||p(\cdot|\theta)) \tag{1}$$

Q-learning

RL problems can be formulated as a Markov Decision Process (MDP) described as a tuple, $M =< \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma >$ where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, respectively, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow I\!R$ is the state transition probability kernel, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow I\!R$ is a reward function, and $\gamma \in [0, 1]$ is a discount factor. The value function is defined as $V^{\pi}(s) = \mathbf{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t)|s_0 = s]$ for all $s \in \mathcal{S}$, the expected value of cumulative future rewards starting at a state s and following a policy π thereafter. The action-value (Q) function is defined as the value for a state-action pair, $Q^{\pi}(s, a) = \mathbf{E}_{\pi}[\sum_{t=0}^{\infty} \gamma^t r_t(s_t, a_t)|s_0 = s, a_0 = a]$ for all $s \in \mathcal{S}, a \in \mathcal{A}$. The objective of a learning agent in RL is to find an optimal policy $\pi^* = \operatorname{argmax}_{\pi} V^{\pi}$. Finding the optimal values, $V^*(\cdot)$ and $Q^*(\cdot, \cdot)$, requires to solving the Bellman optimality equation:

$$\begin{aligned} Q^*(s, a) &= \mathbf{E}_{s' \sim P(\cdot|s,a)}[R(s, a) + \gamma \max_{a' \in \mathcal{A}} Q(s', a')] \\ V^*(s) &= \max_{a \in \mathcal{A}(s)} Q^*(s, a) \quad \forall s \in \mathcal{S} \end{aligned} \tag{2}$$

where s' is the subsequent state after executing the action a at the state s.

Q-learning is the most popular off-policy TD learning technique due to its relatively easy implementation and guarantee of convergence to an optimal policy (Watkins and Dayan 1992; Kaelbling, Littman, and Moore 1996). Q-learning updates an Q-value of the current state and action pair after observing a reward $R(s, a)$ and the next state s' (one-step TD learning). The update is based on *TD error* - a difference between the *TD target*, $R(s, a)+\gamma \max_b Q(s', b)$, and the current $Q(s, a)$ with a learning rate $\alpha \in [0, 1]$ as the below equation.

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left(R(s, a) + \gamma \max_b Q(s', b) - Q(s, a) \right)$$

Bayesian Q-learning with ADF

Belief Updates on Q-values

We define $Q_{s,a}$ as a Gaussian random variable with mean $\mu_{s,a}$ and variance $\sigma^2_{s,a}$ corresponding to the action value function $Q(s, a)$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. We assume that the random variables for different states and actions are independent and have different mean and variance:

$$Q_{s,a} \sim \mathcal{N}(\mu_{s,a}, \sigma^2_{s,a})$$

where $\mu_{s,a} \neq \mu_{s',a'}$ if $s \neq s'$ or $a \neq a'$ $\forall s \in \mathcal{S}, \forall a \in \mathcal{A}$.

According to the Bellman optimality equation in Eq.2, we can define a random variable for $V(s)$ as $V_s = \max_a Q_{s,a}$. We assume that V_s is independent of $\{Q_{s,a}\}_{\forall a \in \mathcal{A}}$ given the related parameters $\{\mu_{s,a}, \sigma^2_{s,a}\}_{\forall a \in \mathcal{A}}$. In general, the maximum of Gaussian random variables, $M = \max_{1 \leq k \leq N} X_k$ where $X_k \sim \mathcal{N}(\mu_k, \sigma^2_k)$ for $1 \leq k \leq N$, has a following distribution:

$$Pr\,(M = x) = \sum_{i=1}^{N} \frac{1}{\sigma_i} \phi\left(\frac{x - \mu_i}{\sigma_i}\right) \prod_{j \neq i}^{N} \Phi\left(\frac{x - \mu_j}{\sigma_j}\right) \tag{3}$$

where $\phi(\cdot)$ is the standard Gaussian probability density function (PDF) and $\Phi(\cdot)$ is the standard Gaussian cumulative distribution function (CDF). Note that Eq.3 is no longer Gaussian.

In the Bayesian perspective of the one-step TD learning, the beliefs on $\mathbf{Q} = \{Q_{s,a}\}_{\forall s \in \mathcal{S}, \forall a \in \mathcal{A}}$ can be updated at time t after observing a reward r_t and the next state s_{t+1} using the Bayes rule. In order to reduce notation, we drop the dependency on t denoting $s_t = s$, $a_t = a$, $s_{t+1} = s'$, $r_t = r$, and also define a causally related 4-tuple $\tau =< s, a, r, s' >$. Since the observation on $Q_{s,a}$ in the one-step TD learning is $r + \gamma V_{s'}$ according to the Bellman optimality equation, the likelihood becomes $p(r + \gamma V_{s'}|\mathbf{q}, \theta) = p_{V_{s'}}((q - r)/\gamma|s', \mathbf{q}, \theta)$ where $\mathbf{q}$ corresponds to $\mathbf{Q}$ and q is a value in $\mathbf{q}$ corresponding to $Q_{s,a}$. θ is a set of mean and variance of $\mathbf{Q}$. From the independence assumptions on $\mathbf{Q}$ and $\{V_s\}_{\forall s \in \mathcal{S}}$, the posterior update can be reduced to an update only for the belief on $Q_{s,a}$:

$$\hat{p}_{Q_{s,a}}(q|\theta, r, s') \propto p_{V_{s'}}\left(\frac{q - r}{\gamma} \middle| q, s', \theta \right) p_{Q_{s,a}}(q|\theta)$$

With the distributions over $V_{s'}$ and $Q_{s,a}$, the resulting posterior distribution is derived as follow (derivation details in the appendix):

$$\begin{aligned} &\hat{p}_{Q_{s,a}}(q|\theta, r, s') \\ &= \frac{1}{Z} \sum_{b \in \mathcal{A}} \frac{c_{\tau,b}}{\bar{\sigma}_{\tau,b}} \phi\left(\frac{q - \bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\right) \prod_{\substack{b' \in \mathcal{A} \\ b' \neq b}} \Phi\left(\frac{q - (r + \gamma\mu_{s',b'})}{\gamma\sigma_{s',b'}}\right) \end{aligned} \tag{4}$$

where Z is a normalization constant and

$$c_{\tau,b} = \frac{1}{\sqrt{\sigma^2_{s,a} + \gamma^2\sigma^2_{s',b}}} \phi\left(\frac{(r + \gamma\mu_{s',b}) - \mu_{s,a}}{\sqrt{\sigma^2_{s,a} + \gamma^2\sigma^2_{s',b}}}\right) \tag{5}$$

$$\bar{\mu}_{\tau,b} = \bar{\sigma}^2_{\tau,b}\Big(\frac{\mu_{s,a}}{\sigma^2_{s,a}} + \frac{r + \gamma\mu_{s',b}}{\gamma^2\sigma^2_{s',b}}\Big) \tag{6}$$

$$\bar{\sigma}^2_{\tau,b} = \Big(\frac{1}{\sigma^2_{s,a}} + \frac{1}{\gamma^2\sigma^2_{s',b}}\Big)^{-1} \tag{7}$$

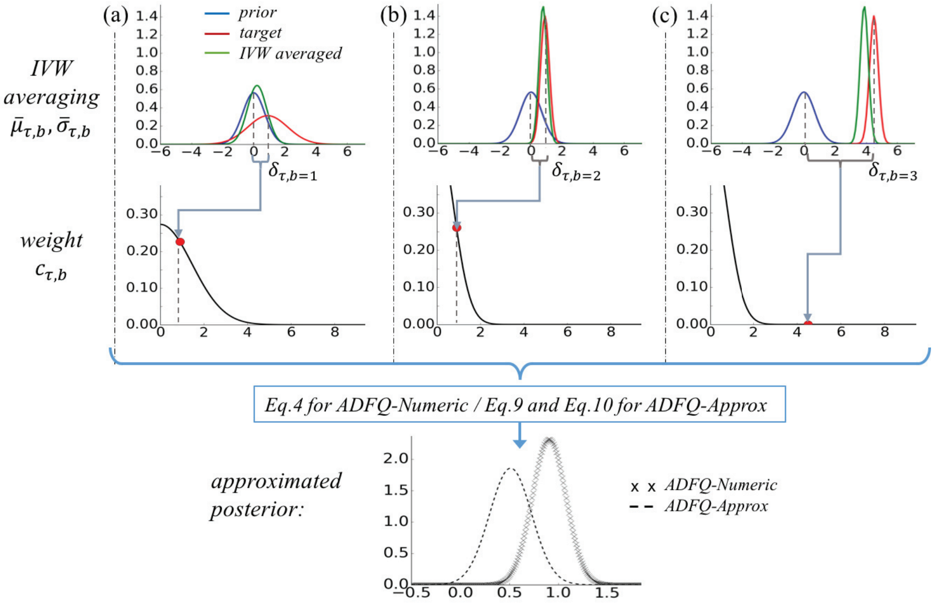


Figure 1: An example of the ADFQ update when $|\mathcal{A}| = 3, r = 0.0, \gamma = 0.9$. The first row illustrates normal distributions with $\bar{\mu}_{\tau,b}, \bar{\sigma}_{\tau,b}$ (green) determined by prior (blue) with $\mu_{s,a} = 0.0, \sigma^2_{s,a} = 0.5$, and a target distribution from each possible action for the next state (red). Numerical values for the subsequent state and action pairs are uncertainty **(a)** $b = 1$: $\mu_{s',b} = 1.0, \sigma^2_b = 2.0$, **(b)** $b = 2$: $\mu_{s',b} = 1.0, \sigma^2_b = 0.1$, **(c)** $b = 3$: $\mu_{s',b} = 5.0, \sigma^2_b = 0.1$. The second row shows that weight value (y value of a red dot) for each action b is determined by TD error ($\delta_{\tau,b}$, x value of the red dot) and uncertainty measures of $Q_{s,a}$ and $Q_{s',b}$ as in Eq.5. Each graph draws a normal distribution with zero mean and variance $\sigma^2_{s,a} + \gamma^2\sigma^2_{s',b}$.

Note that $c_{\tau,b}, \bar{\mu}_{\tau,b}, \bar{\sigma}_{\tau,b}$ are used only to simplify the expression and no additional parameters were introduced. The first two rows in Fig.1 describe an example of how values of $\bar{\mu}_{\tau,b}, \bar{\sigma}_{\tau,b}$ and $c_{\tau,b}$ are determined. Unlike the Q-learning algorithm which considers only a subsequent action resulting the maximum Q-value in the next step ($\max_b Q(s', b)$) in its update, all actions are considered in Eq.4. As found in Eq.5, the TD error, $\delta_{\tau,b} = (r+\gamma\mu_{s',b})-\mu_{s,a}$, is naturally incorporated in the posterior distribution with the form of Gaussian PDF as a weight, and thus a subsequent action which results a smaller TD error contributes to the update more by $c_{\tau,b}$ as compared in the column (b) and (c) in Fig.1. In addition, $\bar{\mu}_{\tau,b}$ is an inverse-variance weighted (IVW) average of the prior mean and the target mean from observations. (Note that we use a term "target" from TD learning contexts for $r + \gamma\mu_{\tau,b}$ and $\gamma^2\sigma^2_{\tau,b}$.) Therefore, the averaged mean is closer to the prior mean if uncertainty of the prior is smaller than that of the target distribution, and vice versa (The column (a) and (b) in Fig.1).

However, the updated posterior distribution is not consistent with the prior distribution. In the next section, we approximate the posterior to a Gaussian distribution using ADF.

Assumed Density Filtering with Q-Beliefs

Applying Eq.1 with the posterior in Eq.4, we minimize $KL(\hat{p}_{Q_{s,a}}||p)$ with respect to mean μ and variance σ^2 where $p = \mathcal{N}(\cdot|\mu, \sigma^2)$. When the parametric family is a spherical Gaussian, it is easily shown that $\mu^* = \mathbf{E}_{\mathbf{q}\sim\hat{p}_{Q_{s,a}}(\cdot)}[\mathbf{q}]$ and $\sigma^{*2} = \mathrm{Var}_{\mathbf{q}\sim\hat{p}_{Q_{s,a}}(\cdot)}[\mathbf{q}]$. Therefore, the approximated posterior will be a Gaussian distribution having the mean and the variance of the true posterior as its mean and variance, respectively.

It is fairly easy to analytically derive the mean and the variance of the true posterior (Eq.4) when $|\mathcal{A}| = 2$. The derivation and the solutions are presented in the appendix. However, to our knowledge, when $|\mathcal{A}| > 2$, solutions become too complicated or are not known. In the next section, we present an approximated ADFQ algorithm which provides analytic solutions for the mean and the variance as well as reduces the algorithmic complexity.

Approximated ADFQ

If the variance of a Gaussian random variable, $X \sim \mathcal{N}(\mu, \sigma^2)$, approaches 0, its CDF and PDF are approximated to a Heaviside step function, $H(\cdot)$ and a dirac delta function, $\delta(\cdot)$, respectively. Suppose that $\sigma_{s,a} \ll 1$ for all $s \in \mathcal{S}$ and $a \in \mathcal{A}$. The product of the Gaussian CDFs in the Eq.4 is approximated to 1 if $q \geq r + \gamma\mu_{s',b'}$ for all $b' \in \mathcal{A}$, $b' \neq b$, and 0 otherwise. However, when $q = \bar{\mu}_{\tau,b}$, we cannot simply apply the approximation since the PDF approaches infinity:

$$\lim_{\substack{\bar{\sigma}_{\tau,b},\sigma_{s',b'}\\ \to 0}} \frac{1}{\bar{\sigma}_{\tau,b}}\phi\Big(\frac{q-\bar{\mu}_{\tau,b}}{\bar{\sigma}_{\tau,b}}\Big)\cdot\prod_{b'\neq b}\Phi\Big(\frac{q-(r+\gamma\mu_{s',b'})}{\gamma\sigma_{s',b'}}\Big)$$
$$= \infty \cdot 0 \neq 0$$

We define a function $f(\cdot)$ which is the approximation of the above equation when the term of the product of the Gaussian

Algorithm 1: ADFQ

```
Initialize $\mu_{s,a}$, $\sigma_{s,a}$ $\forall s \in \mathcal{S}$ and $\forall a \in \mathcal{A}$
for each time step $t$ do
    $a_t \sim \pi^{action}(s_t; \theta_t)$
    Perform the action and observe $r_t$ and $s_{t+1}$
    $\tau =< s_t, a_t, r_t, s_{t+1} >$
    Compute $c_{\tau,b}$, $\bar{\mu}_{\tau,b}$, $\bar{\sigma}_{\tau,b}$ $\forall b \in \mathcal{A}$ from Eq.5 - 7
    Update $\mu_{s_t,a_t}$ and $\sigma_{s_t,a_t}$ using Eq.4 (ADFQ-Numeric)
        or Eq.10 (ADFQ-Approx)
```

Table 1: ADFQ algorithm

CDFs approaches 0 (e.g. $q < r + \gamma\mu_{s',b'}$ for all $b' \neq b$).

$$f(q;\mu,\sigma) = \begin{cases} \frac{1}{\sigma}\phi\Big(\frac{q-\mu}{\sigma}\Big) & \text{for } q \in [\mu-\epsilon, \mu+\epsilon], \epsilon \ll 1 \\ 0 & \text{otherwise} \end{cases}$$

Then, the posterior distribution is approximated to $p_{Q_{s,a}}(q|\theta', r, s') \approx \hat{p}_{Q_{s,a}}(q)$,

$$\hat{p}_{Q_{s,a}}(q) = \frac{1}{Z}\sum_{b\in\mathcal{A}} c_{\tau,b} f(q; \bar{\mu}_{\tau,b}, \bar{\sigma}_{\tau,b}) \text{ for } q \in (-\infty, +\infty) \quad (8)$$

where the normalization factor Z is $\sum_b c_{\tau,b}$. Further details on the approximation can be found in the appendix.

Mean and Variance of the Approximated Posterior

From integrals $\int x\phi(x)dx = -\phi(x) + C$ and $\int x^2\phi(x)dx = -x\phi(x) + \Phi(x) + C$, we obtain the mean and the variance of $\hat{p}_{s,a}(q)$,

$$\mathbf{E}_{q\sim\hat{p}_{Q_{s,a}}(\cdot)}[q] = \frac{\sum_b c_{\tau,b}\bar{\mu}_{\tau,b}}{\sum_b c_{\tau,b}} \quad (9)$$

$$\text{Var}_{q\sim\hat{p}_{Q_{s,a}}(\cdot)}[q] = \frac{\sum_b c_{\tau,b}\bar{\sigma}^2_{\tau,b}}{\sum_b c_{\tau,b}} \quad (10)$$

Interestingly, the mean and variance of the approximated posterior are weighted sums of $\bar{\mu}_{\tau,b}$ and $\bar{\sigma}^2_{\tau,b}$ for all actions in $\mathcal{A}$, respectively. We call the ADFQ algorithm which uses the approximated mean and variance under the small variance assumption as *ADFQ-Approx* and the ADFQ algorithm which numerically computes the mean and the variance from Eq.4 as *ADFQ-Numeric*. We compared true posterior, ADFQ-Numeric posterior, and ADFQ-Approx posterior with different values for the parameters and presented their results in the appendix. In most cases, ADFQ-Numeric posterior approximates true posterior with very small errors. ADFQ-Approx posterior tends to have a smaller mean and larger variance than true as well as ADFQ-Numeric posteriors. The final algorithm is described in Table.1.

Algorithmic Complexity

As shown in Table.1, both ADFQ-Numeric and ADFQ-Approx require computing $c_{\tau,b}, \bar{\mu}_{\tau,b}, \bar{\sigma}_{\tau,b}$ for all $b \in \mathcal{A}$. In ADFQ-Numeric, mean and variance can be computed by Eq.4 with a form of $\sum_b \phi(b)\prod_{b'\neq b}\Phi(b) = (\prod_{b'}\Phi(b'))\cdot\sum_b \phi(b)/\Phi(b)$ resulting that its computational complexity is reduced to $O(m|\mathcal{A}|)$ where m is the number of samples. For ADFQ-Approx, the computational complexity is $O(|\mathcal{A}|)$. The space complexity for both ADFQ-Numeric and ADFQ-Approx is $O(|\mathcal{S}||\mathcal{A}|)$. As a result, ADFQ-Approx algorithm is as efficient as the Q-learning algorithm, and both ADFQ-Numeric and ADFQ-Approx are more efficient than KTD-Q which computational complexity is $O(|\mathcal{S}|^2|\mathcal{A}|^3)$ and space complexity is $O(|\mathcal{S}|^2|\mathcal{A}|^2)$ in finite state and action spaces.

Connection to Q-learning

We can relate this result to the conventional Q-learning since the mean of the posterior is the weighted sum of the prior mean and the target means.

$$\begin{aligned} \mathbf{E}_{s,a}[q] = &\sum_{b\in\mathcal{A}} \frac{c_{\tau,b}}{Z}\frac{\bar{\sigma}^2_{\tau,b}}{\sigma^2_{s,a}}\mu_{s,a} + \frac{c_{b^*}}{Z}\frac{\bar{\sigma}^2_{b^*}}{\gamma^2\sigma^2_{s',b^*}}(r+\gamma\mu_{s',b^*}) \\ &+ \sum_{b\neq b^*}\frac{c_{\tau,b}}{Z}\frac{\bar{\sigma}^2_{\tau,b}}{\gamma^2\sigma^2_{s',b}}(r+\gamma\mu_{s',b}) \end{aligned} \quad (11)$$

where $b^* = \text{argmax}_{\tau,b}\,\mu_{s',b}$. Suppose that $c_{\tau,b} = 0$ for all $b \neq b^*$ and thus the third term of Eq.11 becomes 0. Then we can define $\bar{\alpha}$ which corresponds to the learning rate in the conventional Q-learning as

$$\bar{\alpha} \equiv \frac{\bar{\sigma}^2_{b^*}}{\gamma^2\sigma^2_{s',b^*}} = \Big(1 + \Big(\frac{\gamma\sigma_{s',b^*}}{\sigma_{s,a}}\Big)^2\Big)^{-1}$$

$\bar{\alpha}$ converges to 1 when $\sigma_{s,a} \gg \sigma_{s',b^*}$ and converges to 0 when $\sigma_{s,a} \ll \sigma_{s',b^*}$. Thus, it naturally provides a learning rate - the smaller the variance of the next state (the higher the confidence), the more $Q_{s,a}$ is updated from the target information rather than the current belief. We can therefore think of the Q-learning with a constant learning rate as a special case of ADFQ with very small variance values in which ratios of the variance values are constant. Since the state transitions are decoupled, the ratio cannot be a constant during the learning unless all the variance is not updated but remains constant ($\sigma_{s',b^*}/\sigma_{s,a} = 1, \forall s, s' \in \mathcal{S}$ and $\forall a, b^* \in \mathcal{A}$). Therefore, in this case, the corresponding learning rate is $\alpha = 1/(1+\gamma^2)$. For example, when $\gamma = 0.9$, it gives an identical result to the Q-learning with α = 0.5525.

Experiments

Algorithms

The ADFQ algorithms were tested with three different action policies: *Bayesian Sampling (BS)* which selects $a_t = \text{argmax}_a\, q_{s_t,a}$ where $q_{s_t,a} \sim p_{Q_{s_t,a}}(\cdot|\theta_t)$, *semi-BS* which performs *BS* with a small probability and greedily selects the best action otherwise, and ϵ-*greedy* which randomly selects

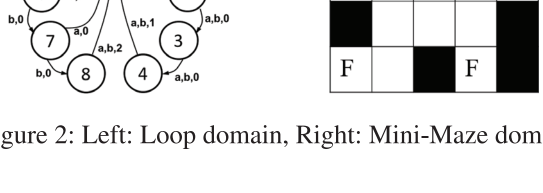


Figure 2: Left: Loop domain, Right: Mini-Maze domain

	Loop	Grid 5x5	Grid 10x10	Mini-Maze
Q-learning, ϵ-greedy	302.4 ± 12.1	150.6 ± 3.8	45.6 ± 3.9	239.7 ± 81.4
Q-learning, Boltzmann	288.2 ± 17.4	61.6 ± 5.5	18.0 ± 1.9	106.1 ± 10.4
ADFQ-Numeric, ϵ-greedy	325.5 ± 13.5	88.5 ± 4.7	21.0 ± 4.9	187.4 ± 92.8
ADFQ-Numeric, semi-BS	329.5 ± 0.8	100.3 ± 5.9	21.3 ± 2.4	220.2 ± 41.1
ADFQ-Numeric, BS	328.4 ± 0.8	116.9 ± 4.0	29.8 ± 3.9	204.7 ± 72.8
ADFQ-Approx, ϵ-greedy	$\mathbf{338.0 \pm 0.0}$	178.1 ± 5.5	$\mathbf{82.7 \pm 5.0}$	$\mathbf{274.8 \pm 80.3}$
ADFQ-Approx, semi-BS	329.2 ± 13.8	$\mathbf{184.7 \pm 4.5}$	80.9 ± 7.1	264.0 ± 67.3
ADFQ-Approx, BS	333.2 ± 3.2	135.9 ± 5.7	51.5 ± 3.3	180.9 ± 47.8
KTD-Q, ϵ-greedy	281.6 ± 5.2	0.6 ± 1.8	0.0 ± 0.0	20.5 ± 16.4
KTD-Q, active learning	157.4 ± 7.4	18.8 ± 2.7	8.0 ± 1.9	55.4 ± 8.6

Table 2: The mean and confidence interval of total cumulative rewards over 10 trials. The number of learning steps are: **Loop** - 1000 steps, **Grid 5x5** - 2000 steps, **Grid 10x10** - 2000 steps, **Mini-Maze** - 5000 steps

an action with ϵ probability, and does it greedily otherwise. For compared algorithms, we experimented Q-learning with ϵ-greedy and Boltzmann action selection rules. We also experimented KTD-Q with ϵ-greedy and with its active learning scheme. The active learning scheme was provided using the variance of the approximated action-value function. We set the initial covariance of the process noise in KTD-Q to be $0I$ and the initial covariance of the observation noise to be 1 as those values were mostly used in the original publication. The scaling factor for the sigma points used in KTD-Q were fixed as 1. For consistency, we used the same method for initializing Q-values, mean parameters in ADFQ $\mu_{s,a} \forall s \in \mathcal{S}, a \in \mathcal{A}$, and the parameter vector in KTD-Q. We assumed that reward values were unknown and initialize the parameters with $r_0/(1-\gamma)$ for non-episodic domains and with r_0 for episodic domains after the first nonzero reward r_0 was observed. The learning started after the initialization. For all algorithms, the discount factor was $\gamma = 0.9$. All other hyperparameters of the experimented algorithms such as initial variance and ϵ were selected through cross-validation and their values are reported in the appendix.

Domains

We tested our algorithms with finite learning steps (T_H) in four different domains which include small/large state space, non-episodic/episodic, and deterministic/stochastic environments: **Loop**($T_H = 1000$) consists of 9 discrete states and 2 actions (a,b). The domain has deterministic state transition. There are +1 reward at state 4 and +2 reward at state 8 as shown in Fig.2. **Grid 5x5** ($T_H = 2000$) is a 2-dimensional 5×5 grid with 25 discrete states and 4 cardinal actions. There is no reward anywhere except at a goal state with +1. The goal state is located opposite to the start state and the agent receives a reward when it reaches the goal state. With a probability 0.1, the learning agent slips in the grid and moves to the right perpendicular direction. **Grid 10x10** ($T_H = 5000$) is similar to Grid 5x5 domain but on a 10×10 grid. In this domain, the agent slips with a probability 0.1 and moves to a randomly chosen perpendicular direction. **Mini-Maze** ($T_H = 5000$) is designed inspired by Dearden's Maze (Dearden, Friedman, and Russell 1998) since the KTD-Q algorithm was not able to handle the Dearden's Maze domain in reasonable computational time. The diagram of Mini-Maze is shown in Fig.2. It has a total of 112 states and 4 cardinal actions. ”S” is the start state and ”G” is the goal state. Three flags are located in ”F” and the agent's goal is to collect the flags and escape the maze through the goal state. It receives a reward equivalent to the number of flags it has collected at the goal state. The Black blocks represent wall and the agent stays at the current state if it performs an action toward a wall. As same as the grid domains, the agent slips with a probability 0.1.

Results

Each algorithm with different action policies was tested 10 times on each domain, and their results were averaged. The sum of rewards ($\sum_{t=0,\cdots,T_H} r_t$) obtained during learning is displayed in Table.2. Learning was paused at every $T_H/100$ step and the current policy was semi-greedily evaluated (used ϵ-greedy with $\epsilon = 0.1$). In the evaluation, the maximum number of steps is bounded by $T_H/50$, and for the episodic domains, it is also terminated when a goal state is reached. Each evaluation was averaged over 10 trials, and the results are shown in Fig.3. For simplification, evaluation results of ADFQ-Numeric are reported in the appendix. Overall ADFQ-Approx outperformed all other algorithms including ADFQ-Numeric.

ADFQ-Numeric performs worse than ADFQ-Approx. This is because the mean of the maximum of Gaussian random variables is equal to or larger than the maximum of means of Gaussian random variables (i.e. $\mathbf{E}[M = \max_{i=1\cdots N} X_i] \geq \max_{i=1\cdots N} \mathbf{E}[X_i]$) which applies to the likelihood distribution, $p(r + \gamma V_{s'}|\mathbf{q}, \theta)$. Thus, the posterior mean can be overestimated and the overestimation amount depends on the discount factor, γ, and variance of $Q_{s',b} \forall b \in \mathcal{A}$. Simple examples showing the dependencies are presented in the appendix. We also experimented ADFQ-Numeric and ADFQ-Approx with $\gamma = 0.5$ in the same domains, and ADFQ-Numeric showed similar performance with ADFQ-Approx (see the appendix).

ADFQ-Approx with the ϵ-greedy action policy resulted in the highest total rewards in three of four domains, but ADFQ-Approx with the semi-BS policy showed similar performances to the semi-greedy evaluation. In the Loop domain, ADFQ-Approx with BS worked the most smoothly and converged to the maximum possible cumulative reward

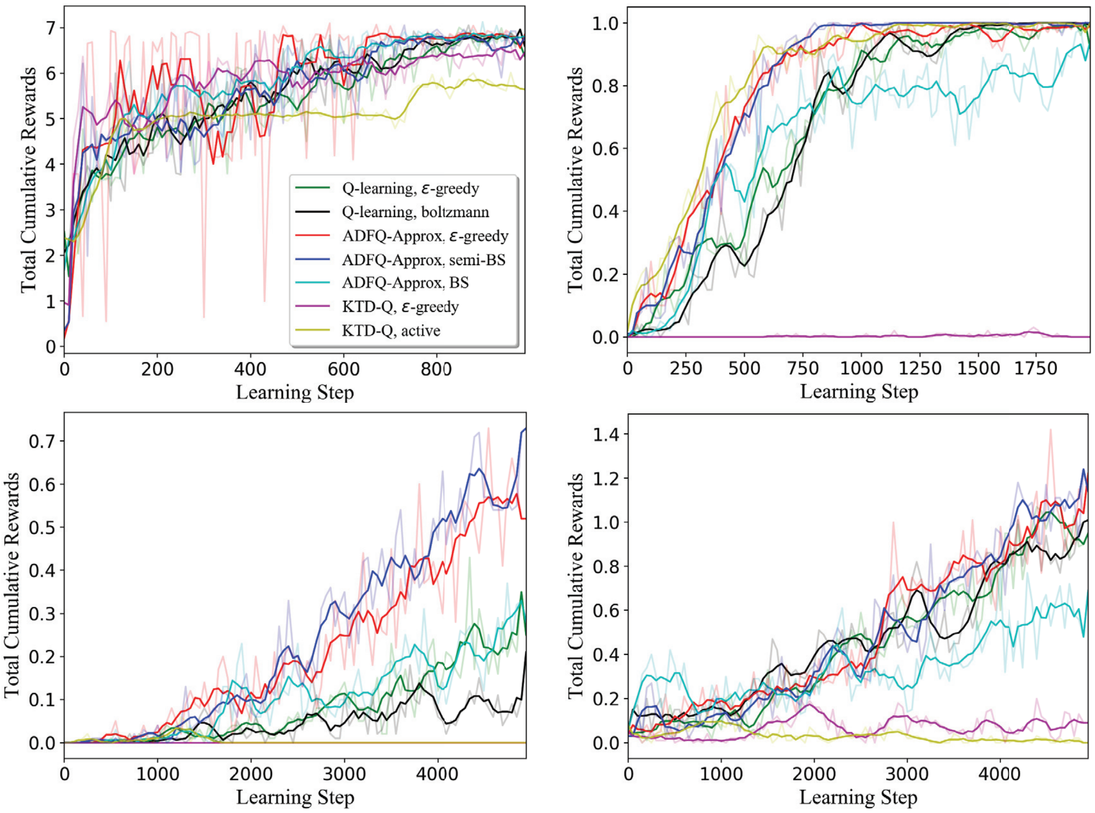


Figure 3: Cumulative rewards in semi-greedy evaluation during learning at each domain, averaged over 10 trials for each algorithm with an action selection rule. The curves were smoothed by a simple moving average with window size 4. From top to bottom, Loop, Grid 5x5, Grid 10x10, Mini-Maze

quickly. However, it didn't perform well in the stochastic domains. KTD-Q didn't work well overall, especially in stochastic and large domains.

Discussion

We proposed an approach to Bayesian off-policy TD method called ADFQ and its two variants, *ADFQ-Numeric* and *ADFQ-Approx*. ADFQ-Approx surpassed the performance of ADFQ-Numeric, Q-learning and KTD-Q in various task domains. With a smaller discount factor, ADFQ-Numeric showed similar performances compared to ADFQ-Approx outperforming other comparing algorithms. The presented ADFQ algorithms demonstrate several intriguing results.

First, unlike the Q-learning algorithm, the ADFQ algorithms incorporate the information of all possible actions for the next state in the update with weights depending on TD errors and uncertainty measures (Eq.4 and Eq.10). As mentioned previously, this provides intuitive update - a state-action pair with higher uncertainty in its Q belief has a smaller weight contributing less to the update. Therefore, we made use of our uncertainty measuress in the value update with natural regularization based on the current beliefs. Second, we were able to connect ADFQ-Approx algorithm to Q-learning and showed Q-learning could be a special case of our algorithm. Third, one of major drawbacks of BRL approaches is that most algorithms are computationally more demanding than standard RL algorithms (Ghavamzadeh et al. 2015). However, ADFQ-Approx is computationally as efficient as Q-learning and both ADFQ algorithms are more efficient than KTD-Q. Fourth, we did not make any deterministic environment assumption and ADFQ worked well on stochastic environments. Lastly, the ADFQ algorithms require only two hyperparameters, initial variance and the discount factor, while other BRL algorithms tend to require many hyperparameters to be chosen.

There are several limitations in the ADFQ algorithms. Convergence analysis is not provided in this paper. This can be achieved in a similar manner to Q-learning since the two algorithms share a resemblance. In addition, as an initial approach, we started from finite state and action spaces. We are extending our method to parameters of function approximation in order to apply the method to a real world example. Lastly, performance of ADFQ-Numeric is dependent on relative numerical values of its initial mean, initial variance, reward/s, and the discount factor. Finding a fundamental rule for setting the values would be a future work.

References

Boyen, X., and Koller, D. 1998. Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth con-*

ference on Uncertainty in artificial intelligence.

Chowdhary, G.; Liu, M.; Grande, R.; Walsh, T.; How, J.; and Carin, L. 2014. Off-policy reinforcement learning with gaussian process. *IEEE/CAA Journal of Automatica Sinica* 1(3):227–238.

Dearden, R.; Friedman, N.; and Andre, D. 1999. Model based bayesian exploration. In *Proceedings of the 15th conference on Uncertainty in artificial intelligence*, 150–159. Morgan Kaufmann Publishers Inc.

Dearden, R.; Friedman, N.; and Russell, S. 1998. Bayesian q-learning. In *AAAI/IAAI*, 761–768.

Duff, M. 2002. Optimal learning: Computational procedures for bayes-adaptive markov decision processes. *PhD thesis, University of Massachusetts, Amherst*.

Engel, Y.; Mannor, S.; and Meir, R. 2003. Bayes meets bellman: The gaussian process approach to temporal difference learning. In *Proceedings of the 20th International Conference on Machine Learning*, volume 20.

Engel, Y.; Mannor, S.; and Meir, R. 2005. Reinforcement learning with gaussian processes. In *Proceedings of the 22nd International Conference on Machine Learning*, 201–208.

Geist, M., and Olivier, P. 2010. Kalman temporal differences. *Journal of artificial intelligence research* 39:483–532.

Ghavamzadeh, M., and Engel, Y. 2006. Bayesian policy gradient algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, 457–464.

Ghavamzadeh, M.; Mannor, S.; Pineau, J.; and Tamar, A. 2015. Bayesian reinforcement learning: A survey. *Foundation and Trends in Machine Learning* 8(5-6):359–483.

Guez, A.; Silver, D.; and Dayan, P. 2012. Efficient bayes-adaptive reinforcement learning using sample-based search. In *Advances in Neural Information Processing Systems (NIPS)*, 1071–1079.

Kaelbling, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4:237–285.

Maybeck, P. S. 1982. Stochastic models, estimation and control. *Academic Press* chapter 12.7.

Opper, M. 1999. A bayesian approach to online learning. *On-Line Learning in Neural Networks*.

Poupart, P.; Vlassis, N.; Hoey, J.; and Regan, K. 2006. An analytic solution to discrete bayesian reinforcement learning. In *Proceedings of the 23rd International Conference on Machine Learning*, volume 20, 697–704.

Strens, M. 2000. A bayesian framework for reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, 943–950.

Watkins, C. J., and Dayan, P. 1992. Q-learning. In *Machine Learning*, 279–292.

State Abstraction Synthesis for Discrete Models of Continuous Domains

Jacob Menashe, Peter Stone
{jmenashe,pstone}@cs.utexas.edu
The University of Texas at Austin
Austin, TX USA

Abstract

Reinforcement Learning (RL) is a paradigm for enabling autonomous learning wherein rewards are used to influence an agent's action choices in various states. As the number of states and actions available to an agent increases, so it becomes increasingly difficult for the agent to quickly learn the optimal action for any given state. One approach to mitigating the detrimental effects of large state spaces is to represent collections of states together as encompassing "abstract states".

State abstraction itself leads to a host of new challenges for an agent. One such challenge is that of automatically identifying new abstractions that balance generality and specificity; the agent must identify both the similarities and the differences between states that are relevant to its goals, while ignoring unnecessary details that would otherwise hinder the agent's progress. We call this problem of identifying useful abstract states the Abstraction Synthesis Problem (ASP).

State abstractions can provide a significant benefit to model-based agents by simplifying their models. T-UCT, a hierarchical model-learning algorithm for discrete, factored domains, is one such method that leverages state abstractions to quickly learn and control an agent's environment. Such abstractions play a pivotal role in the success of T-UCT; however, T-UCT's solution to ASP requires a fully discrete state space.

In this work we develop and compare enhancements to T-UCT that relax its assumption of discreteness. We focus on solving ASP in domains with multidimensional, continuous state factors, using only the T-UCT agent's limited experience histories and minimal knowledge of the domain's structure. Finally, we present a new abstraction synthesis algorithm, RCAST, and compare this algorithm to existing approaches in the literature. We provide the algorithmic details of RCAST and its subroutines, and we show that RCAST outperforms earlier approaches to ASP by enabling T-UCT to accumulate significantly greater total reward with minimal expert configuration and processing time.

1 Introduction

The efficiency with which an AI agent learns the dynamics of a domain is heavily influenced by that agent's internal representation of the world. This is especially true for complex, multi-faceted domains that are often found in the real world. For this reason, AI researchers are often forced to wrestle with the so-called *curse of dimensionality* wherein the complexity of a domain scales exponentially with the number of variables used to describe it.

A traditional solution to this problem is through the use of hierarchical layers of abstraction; rather than painstakingly learning about every individual state possible in some domain, an agent can instead group large numbers of states together and consider only this abstract representation during the learning process. While such representations are merely helpful for simpler domains, abstractions are essentially a requirement when applying any form of machine learning to domains over real-valued variables (such as one's position in continuous space).

The T-UCT algorithm (Menashe and Stone 2015) exemplifies the success of this approach by learning decision-tree-based models from scratch in hierarchically structured domains. However, T-UCT has no internal mechanism for modeling continuous state, and while it is designed for domains with factored state representations, it performs poorly on domains whose state factors span large value spaces. When faced with value spaces of infinite cardinality, T-UCT often performs worse than chance.

In this work we augment T-UCT with mechanisms for efficiently modeling actions and state transition dynamics in continuous, factored state spaces. We call this augmented version of T-UCT *Continuous* T-UCT (CT-UCT). We design our CT-UCT augmentations in such a way that a variety of competing abstraction synthesis algorithms can be "plugged in" and evaluated on a single hierarchical, continuous learning task, toward the ultimate goal of enabling a CT-UCT agent to maximize its total accumulated extrinsic reward.

In Section 2 we describe the necessary background for T-UCT and CT-UCT as well as the past research on abstraction synthesis. The primary challenge in developing CT-UCT is that of retrofitting T-UCT's discrete model-learning infrastructure with the necessary machinery for learning abstractions over continuous space; in Section 3 we present a novel abstraction synthesis algorithm, the *Recursive Cluster-based Abstraction Synthesis Technique* (RCAST), which achieves this feat of identifying abstractions that can be consumed by CT-UCT's modeling framework.

In Section 4, we compare RCAST with alternative algorithms from existing literature by plugging these algorithms into CT-UCT and evaluating their performance on a challenging HRL task. Finally in Section 5 we conclude and discuss future work.

2 Background and Related Work

In this paper we focus our discussion on state abstraction synthesis in the context of Reinforcement Learning (RL). Thus we begin our discussion of background literature with a brief overview of model-based RL, and then proceed to cover the related work on state abstraction.

2.1 Model-based Reinforcement Learning

Model-based reinforcement learning is a branch of reinforcement learning in which an agent uses a model to predict the effects of its actions in the environment. In effect, while classical Reinforcement Learning is concerned with learning a value function dependent upon R, model-based reinforcement learning is additionally concerned with invoking (and possibly learning) an approximation of P. Often, the agent's model of P is used as an intermediate step toward improving the value function.

In our work we consider models based on Conditional Probability Trees (CPTs), which are a form of decision tree in which each internal node "splits" based on the values of a particular state factor F. Each branch from such an internal node denotes a value (or set of values) for F. (Jonsson and Barto 2007) describe how such a model can encode the dynamics of a particular RL domain and be used to predict a state s_t from its predecessor s_{t-1} and an action a_t taken in s_{t-1} for some timestep t.

(Vigorito and Barto 2010) describe the VISA algorithm, which uses CPTs and $\langle s_t, a_t, s_{t+1}\rangle$ histories to learn a model of its environment from scratch. (Menashe and Stone 2015) use T-UCT (based on the UCT algorithm (Kocsis and Szepesvári 2006)) with the CPT framework of (Jonsson and Barto 2007) to improve learning performance and sample efficiency in comparison with VISA, however both T-UCT and VISA assume discrete MDPs. In our work, RCAST provides the key mechanism for relaxing this assumption of discreteness by producing discrete state abstractions over continuous value spaces.

2.2 State Abstraction Synthesis

Small, finite, and factored state spaces give rise to useful and intuitively defined state abstraction mechanisms. (Jong and Stone 2005) propose a method for state abstraction in such factored state spaces through identification of so-called "irrelevant" factors. For instance, if the state space S has factors X and Y, then an abstract state might be a particular assignment $X = x_0$ with no assignment for Y. In this case, the abstract state space S' consists of $|X|$ abstract states each encompassing $|Y|$ primitive states. (Jonsson and Barto 2005) use decision tree models of the state space toward a similar end, where each branch encodes an assignment of variables to values, and omitted variables represent those that are irrelevant for a particular action model.

Rather than defining abstractions in terms of critical values, there has been ample work on defining abstractions in terms of their relevance to "macro" actions using the options framework (Sutton, Precup, and Singh 1999). "Bottleneck" options are one such example where the state space is divided into regions on either side of highly-traversed intermediate states ("bottleneck" states). The initiation and termination sets of such options each designate two distinct abstract states that can be used for planning in lieu of the primitive state space (Menache, Mannor, and Shimkin 2002; McGovern and Barto 2001; Stolle and Precup 2002). Macro actions give rise to Hierarchical Reinforcement Learning (HRL), where a single macro action may consist of many sub-actions, and may itself comprise part of a more general macro action. State abstraction is often a central component of an HRL algorithm; T-UCT is no exception, as its entire model-learning framework is concerned with identifying dependencies between abstract states.

State abstractions can be more difficult to synthesize in domains with continuous-valued state variables. Due to the negligible likelihood of visiting a single real-valued state multiple times, an agent must instead attempt to visit the neighborhoods about such values for effective planning. Planning with neighborhoods raises the challenge of determining the appropriate size and shape of such neighborhoods. Option-based state abstraction extends naturally into continuous domains, however this does not relieve the aforementioned difficulty in identifying continuous neighborhoods. Such neighborhoods can be classified using a traditional supervised learning approach (Konidaris and Barto 2009), but this relies on large numbers of sample trajectories and predefined classes used to label the samples.

Iterative Half-Space (IHS) Abstraction Synthesis Many alternative approaches to abstraction synthesis rely on iteratively dividing the state space into half-spaces using hyperplanes (Jonsson and Barto 2001; Quinlan and others 1992; Liu, Xia, and Yu 2000; Kohavi 1996; Fayyad and Irani 1993). Such iterative half-space (IHS) approaches identify optimal hyperplanes for splitting some space one at a time until all of the splits necessary to fully describe the space's dynamics have been identified. While this technique can achieve arbitrary levels of precision, it invariably results in creating unnecessary abstract states as a side-effect of the iterative halving process. Moreover, when multiple half-spaces are required for meaningful separation of datapoints, the initial splits must be performed with limited statistical indication of their relevance. Thus the algorithm must split aggressively in anticipation of high quality abstractions many iterations in the future, and at the same time split conservatively to avoid creating abstractions that are harmful to the learning process. The overall effect is that such algorithms tend to be either sample-inefficient or inaccurate.

(Fayyad and Irani 1993) tackle the state abstraction problem with unidimensional continuous factors by hierarchically splitting continuous intervals into two parts at a time, but even this approach suffers from creating unnecessary abstract states and is poorly suited to continuous factors over multiple dimensions.

Our work improves upon that of Fayyad and Irani by both removing the need for creating unnecessary abstract states and enabling abstraction over continuous factors of arbitrary dimensionality. Section 3 provides a more detailed description of these differences.

kd-tree Discretization k-Dimensional Trees, originally described by (Friedman, Bentley, and Finkel 1977) provide an effective means of partitioning continuous state with discrete and succinctly specifiable bounds to arbitrary levels of precision. Since each facet of a kd-tree is a hyperplane, any kd-tree is equal to some combination of half-space bounds, and thus the argument can be made that a kd-tree partitioning can be inferred via IHS abstraction synthesis. However, such inferrences may not always be possible within reasonable bounds on processing time or computational complexity; an algorithm which can identify kd-tree partitions may therefore outperform an IHS algorithm in time-bound domains.

The Parti-Game Algorithm (Moore 1994) is an example of how kd-tree-based abstractions can be beneficial in representing discrete decision boundaries over continuous-valued state spaces of arbitrary dimensionality. This algorithm is designed for deterministic goal-oriented RL problems where individual leaves are mapped to decisions; however, its core idea of representing decision boundaries with kd-trees shows promise in the more open-ended abstraction synthesis problem.

(Reynolds 2000) extend the application of kd-trees to more traditional RL problems with the Variable Resolution Model-Free Function Approximation Algorithm. Here Reynolds shows that kd-trees can be used to approximate action dynamics over continuous domains without the need for deterministic state transitions or predefined goal states.

While kd-trees are used extensively in past work for the purpose of modeling dynamics or representing decision trees, we know of no other work where kd-trees are applied to the Abstraction Synthesis Problem in the manner we describe below. In Section 3 we introduce our own solution to the abstraction synthesis problem, where we apply kd-trees to the task of partitioning continuous, multidimensional state abstractions.

3 The RCAST Algorithm

This section introduces RCAST, one of the primary contributions of this work and the key to enabling CT-UCT to scale to continuous state spaces. We will first visually depict RCAST on a hypothetical dataset in Section 3.1, and then describe an implementation of RCAST in Section 3.2.

At each timestep $t-1$ in an MDP, an RL agent chooses an action a to take in state s_t, and then experiences the resulting state s_{t+1}. Thus, an agent which keeps track of these $\langle s_t, a_t, s_{t+1}\rangle$ tuples can analyze them to predict future experiences. A key feature of an RL model is the ability to predict s_{t+1} from s_t and a, and in a factored domain an agent may wish to specifically predict the value of some output factor F_o in s_{t+1} given s_t and the operative action a. If F_o is causally related to some other factor F_i, then the value of F_i in s may be of particular relevance to predicting F_o.

The CPT framework used by VISA (Jonsson and Barto 2006) and T-UCT (Menashe and Stone 2015) enables such factor-specific predictions. A CPT allows an agent to predict the value of F_o *given* the action a and the value of F_i, but both of these algorithms assume that F_i and F_o take on discrete values, and that an agent can keep track of *all possible values* of F_i when predicting how a will alter F_o. CT-UCT relaxes this assumption and allows an agent to discretely model F_i and F_o even when they take on continuous and multidimensional values by invoking an abstraction synthesizer, namely RCAST; RCAST's role is thus to identify useful abstractions over F_i's value space, so that they may be used for branching decision trees in the CPT framework of (Jonsson and Barto 2007).

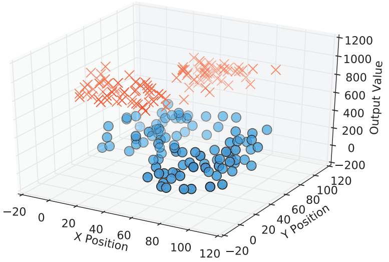


Figure 3.1: A hypothetical dataset D consisting of inputs on the xy plane and output on the z axis.

3.1 Visual Example

Before we describe the algorithmic details of RCAST we will begin by visually depicting RCAST using a hypothetical experience history H of $\langle s_t, a_t, s_{t+1}\rangle$ tuples for an RCAST agent in some domain. Assume that we are interested in understanding whether some state factor F_o depends on another F_i. For ease of visualization let us assume that $\dim(F_o) = 1$ and $\dim(F_i) = 2$.

Before analyzing the interaction between these two factors we first project H into an $\mathbb{R}^n$ subspace where $n = \dim(F_i) + \dim(F_o) = 3$. This projection $P : (S \times A \times S) \to \mathbb{R}^3$ maps $\langle s_t, a_t, s_{t+1}\rangle$ to (x, y, z) where (x, y) is the value of factor F_i in state s_{t-1} and z is the value of factor F_o in state s_t. We denote the image $P(H) = \{(x, y, z)\}$ as D.

In Figure 3.1 we see a scatter plot representation of D. Identifying a dependence relationship from F_i to F_o is therefore similar to the task of predicting z from (x, y). The shapes and colors used in Figure 3.1, as well as all figures in Section 3.1, are not available to the algorithm and are shown strictly for ease of visualization.

Figures 3.3 and 3.2 show the same dataset D restricted to F_i and F_o, respectively. From Figure 3.2 it is clear that two distinct classes of data exist; however, our goal is not to simply identify these classes, but also to use them for classifying tuples in F_i. Thus we wish to partition the plot in Figure 3.3 such that its projection into F_o's value space also partitions the data in Figure 3.2 according to the two obvious classes.

RCAST creates this partitioning by first clustering datapoints in the full $F_i + F_o$ value space (Figure 3.1) and then using these clusters to create a labeled kd-tree in the F_i value

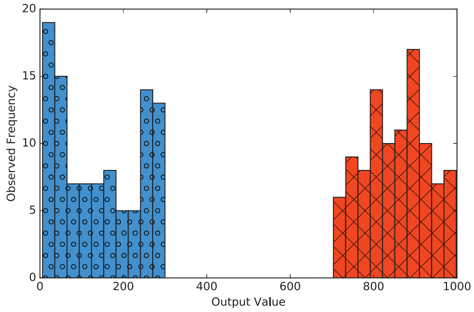


Figure 3.2: A histogram of the output (z) values in the dataset D from Figure 3.1.

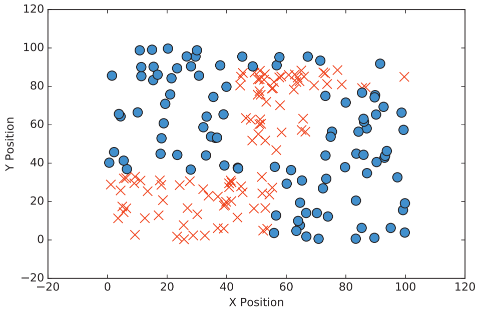


Figure 3.3: A view of D from Figure 3.1 projected onto the input (xy) plane.

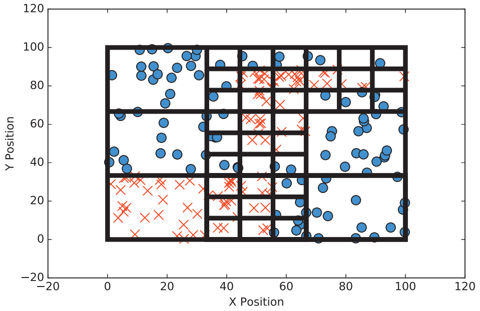


Figure 3.4: A kd-tree T which partitions the dataset D based on its classes of output (z) values.

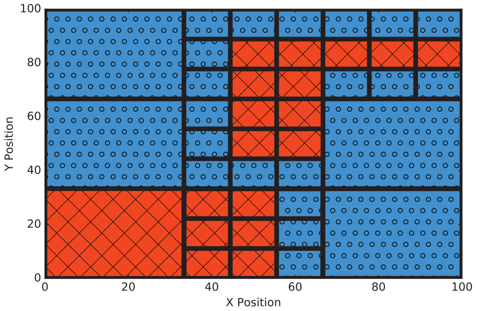


Figure 3.5: A kd-tree T with filled regions visually depicting the labels applied to its leaves.

space (Figure 3.3). The labeled tree describes a partition of the F_i value space, enabling an agent to map from points in F_i to clusters in F_o.

Figure 3.4 shows the kd-tree T generated for D. In most areas the tree is only one layer deep, however the variety of points found in some regions of the value space necessitate secondary levels of refinement. Here we use a discretization factor of $\delta = 3$, resulting in $3^{\dim(F_i)} = 3^2 = 9$ subdivisions at each level, however we note that δ is configurable in the general case.

The fully labeled T in Figure 3.5 is a classifier that maps F_i-value coordinates to labels. The union of the regions encompassed by the leaves of T for some label l can therefore be considered an abstract state over F_i which is relevant to predicting F_o. We can divide T according to these labels, creating a set of abstract meta-states which can then be integrated into a discrete model. Similar to the way in which singleton values can partition a tabular space for a discrete CPT model, these abstractions partition a continuous space for the same purpose (see (Jonsson and Barto 2007)). In this way we are able to discretely model F_o's dependence upon F_i even though these factors describe multidimensional continuous values.

3.2 Algorithm Description

RCAST's purpose is to analyze observed dynamics in a given environment and identify key areas of the environment that exhibit similar dynamics. The areas identified by RCAST then inform model refinements which allow an agent to use its experiences to knowledgeably plan its actions.

Algorithm 3.1a provides pseudo-code to describe RCAST. Line 1 defines the basic inputs to the algorithm including an *input factor* F_i, an *output factor* F_o, a dataset D, and an orthotope Q describing the bounds of F_i. In calling this function we assume that changes in F_o depend on F_i when F_i's value falls within Q. RCAST analyzes D to identify the specifics of this relationship, returning a set of subspaces of Q which serve as abstractions over the value space of F_i.

In Line 2 we see that the use of the $\sqcap$ operator applied to D and Q. This operator restricts the dataset to those datapoints whose predecessor states' value assignments for F_i fall within the bounds of Q. Intuitively, this means that each state-action-state sequence “started" in Q. Line 3 then clusters this subset of points using Expectation-Maximization Clustering, which produces as many clusters as necessary to maximize the BIC score of successive Expectation Maximization iterations. We use Expectation Maximization over Gaussian models (implemented with OpenCV (Bradski 2000)).

Line 4 projects the clusters' datapoints into the value space Q of F_i so that Q can be partitioned in accordance with these clusters. Line 5 then hierarchically partitions Q according to C' using Algorithm 3.1b as the partitioning subroutine. Algorithm 3.1b produces a kd-tree with leaves labeled according to the clusters they encompass. In Line 6 this tree is

```
 1: function RCAST($F_i$, $F_o$, $D$, $Q$)
 2:     $D' \leftarrow D \sqcap Q$
 3:     $C \leftarrow \mathrm{EM}(D')$
 4:     $C' \leftarrow F_i(C)$
 5:     $T \leftarrow$ H-Part$(Q, C')$
 6:     $A \leftarrow \mathrm{SplitLabels}(T)$
 7:     return $A$
 8: end function
```

(a) An implementation of RCAST.

```
 1: function H-PART($Q$, $C$)
 2:     $T \leftarrow$ an empty $k$d-tree with label $\varnothing$
 3:     $C' \leftarrow \{c \sqcap Q | c \in C, c \sqcap Q \neq \emptyset\}$
 4:     if $|C'| = 1$ then
 5:         $T.label \leftarrow c.label$ where $c \in C'$
 6:     else if diameter($Q$) $\leq 1$ then
 7:         $T.label \leftarrow$“multi"
 8:     else
 9:         $d \leftarrow$ dimensionality of $Q$
10:         $\mathcal{Q} \leftarrow$ Partition $Q$ into $\delta^d$ equal parts
11:         for $q \in \mathcal{Q}$ do
12:             $T_q \leftarrow$ H-Part$(q, C)$
13:             $T.children \leftarrow T.children \cup \{T_q\}$
14:         end for
15:     end if
16:     return $T$
17: end function
```

(b) H-Part: A hierarchical partitioning algorithm.

Algorithm 3.1: RCAST and its primary subroutine H-Part

partitioned according to its labeling, with each subtree containing all the leaves for a particular label. Since each of these leaves defines an orthotopic subspace of Q, each subtree is associated with a distinct subspace of Q, namely the union of its leaves' orthotopes. Each such union is an abstraction over the value space of F_i.

Algorithm 3.1b describes a general method for generating a kd-tree from a set of clusters C and an encompassing value space Q. In Line 3 the algorithm restricts its cluster set C to only those clusters with datapoints in Q. Next, H-Parttakes one of three actions. If D contains only one class of datapoint, the current subtree becomes a single leaf with its label taken from that class (Line 5). If D contains multiple classes of datapoints but Q is of minimal diameter[1] then T is designated as having the special label “multi" (Line 7). Otherwise, if the diameter of Q is large enough, the algoritm subdivides and recurses (Line 12). Ultimately the algorithm labels every leaf with either a class label or “multi". We use a static minimum diameter of 1 in this work.

In the context of CPTs, the partitions produced by Algorithm 3.1a are used to split leaves of the CPT. (Jonsson and Barto 2005) show that a CPT leaf can be split along some input factor F_i' by creating one child leaf per value of that factor. In a sense, the discrete value space of F_i' is partitioned into singleton subspaces where each value of F_i comprises one subspace. In the context of a continuous and multidimensional F_i, singleton partitioning is not possible, so instead Algorithm 3.1b produces the aforementioned unions of orthotopes to represent arbitrary subspaces of F_i's value space, with each union corresponding to a single leaf added to the CPT.

Thus, we can refine a CPT model over a continuous, multidimensional F_i by partitioning its value space and then creating one new CPT leaf per label. In Section 4 we evaluate this approach empirically and compare against alternative abstraction synthesis methods.

4 Experiments

This section describes a set of experiments performed to evaluate the effectiveness of RCAST in comparison to both IHS and a deep regression network (DRN). While it would not be possible to evaluate RCAST against all the alternative approaches to abstraction synthesis described in Section 2, IHS and DRN are representative of the state of the art and serve as reasonable proxies for most other methods. Implementation details cannot be included due to space constraints.

4.1 The Continuous Taxi Domain (C-Taxi)

In our work we are interested in evaluating T-UCT and its derivatives on a HRL task over a factored, continuous state space. The Taxi domain (Hengst 2002) is a common choice for evaluating HRL algorithms, but lacks the property of having a continuous-valued state space. We therefore employ a modified version of Taxi in our work, which we refer to as Continuous Taxi (C-Taxi). Rather than the traditional 5-by-5 discrete grid, we use a 100-by-100 continuous grid containing multiple rectangular regions in which anywhere from 1 to 1000 passengers may be picked up by the Taxi agent. More formally, the state space of this domain is the set of 3-tuples $\langle x, y, k\rangle$ where x, y is the position of the agent in the grid and k is the number of passengers currently held by the agent. There are 6 actions including actions for movement in the four cardinal directions N, S, E, W as well as the pickup and dropoff actions P, D. When a movement action is executed the agent is transported a uniformly random distance between 5 and 10 units in the appropriate direction. When P is executed in a pickup region, a uniformly random number of between 500 and 1000 passengers is picked up. Dropoffs always unload all passengers.

The “goal" of the C-Taxi domain is to drop off passengers quickly; thus, the agent receives a reward of -1 for any movement actions, and a reward of 0 for pickups. When the agent executes the dropoff action D it receives a reward equal to the number of passengers that were dropped off.

A perfect abstraction over this domain (with respect to the pickup action and the vehicle's passenger count) would partition the grid into two distinct regions: one non-contiguous region matching the union of the pickup regions, and a second region perfectly complementing the first. When used as the basis for splitting a decision tree, such abstractions would allow an agent to reason with respect to perfect knowledge of where pickups occur, and where they don't.

[1] Here the *diameter* of an orthotope O is defined as $\sup\{d(x, y)|x, y \in O\}$ where d is Euclidean distance.

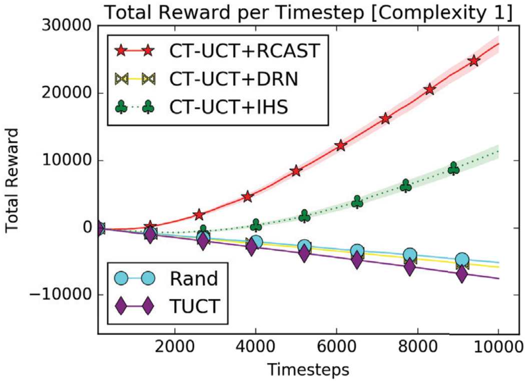


Figure 4.1: A comparison of the CT-UCT+IHS, CT-UCT+DRN, T-UCT, Random, and CT-UCT+RCAST algorithms over many C-Taxi domain instances.

4.2 Experiment Setup

In each experiment we evaluate a selection of HRL algorithms on the task of accumulating extrinsic reward in the C-Taxi domain. Since T-UCT is a model-learning algorithm that relies on extrinsic reward, we modify its internal exploration selection mechanism such that it decides between exploitation-oriented and exploration-oriented targets. During its target selection phase (see Section 3.2 of (Menashe and Stone 2015)), the T-UCT algorithm selects a target context based on its expectation of earned intrinsic reward. For the purpose of evaluating T-UCT on its ability to earn extrinsic reward, we modify this process such that targets are randomly selected based on expected *extrinsic* reward. This allows T-UCT and its derivatives to exploit their learned models with respect to the domain's reward signal. This modification is applied to all such algorithms in this work.

We present empirical results on total reward earned in the C-Taxi domain for the following algorithms: Random (uniformly random action selection), T-UCT[2], CT-UCT+IHS, CT-UCT+DRN, and CT-UCT+RCAST. The following figures show the average results over different levels of complexity. A domain of complexity n contains n pickup regions and n dropoff regions. We provide results for complexities 1, 2, and 3 below. Within each complexity level, we take the average of the results obtained over 30 different C-Taxi instances, each having differing random placements and sizes of pickup and dropoff regions. We then perform 10 evaluations per agent on each such instance, and record each agent's total processing time and accumulated reward every 100 timesteps.

4.3 Results

Experimental results are shown in Figures 4.1 and 4.2. In both figures, "Complexity n" indicates that n pickup regions and n dropoff regions are generated for every instance of the C-Taxi domain. In both result sets, values are averaged over

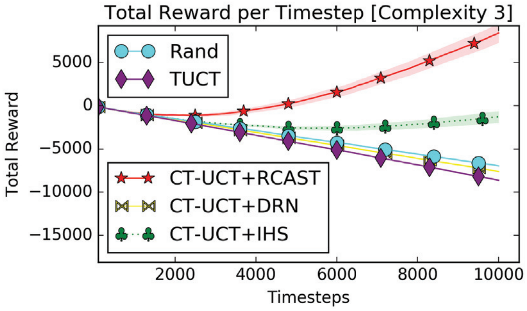


Figure 4.2: A comparison of the CT-UCT+IHS, CT-UCT+DRN, T-UCT, Random, and CT-UCT+RCAST algorithms over many C-Taxi domain instances.

30 domain instances with 10 trials per algorithm per domain instance. Shaded regions represent standard error.

In Figure 4.1 we see that RCAST significantly outperforms every other algorithm ($p \ll .001$). Figure 4.2 indicates that as complexity increases the performance gap only widens. These results show that RCAST is able to efficiently handle complex abstraction synthesis problems and allow for efficient exploration and exploitation in these domains.

5 Conclusion and Future Work

In this work we have described RCAST, a new method for synthesizing abstract states based on observed data. We have used RCAST as the core abstraction synthesis mechanism of CT-UCT, and thereby enabled T-UCT to produce state abstractions for continuous space and effectively incorporate these abstractions into its discrete decision tree model. Moreover, we have shown that RCAST is superior to alternative approaches to abstraction synthesis with respect to total accumulated extrinsic reward, and is competitive to alternatives with respect to time and configuration complexity.

Another interesting focus for future work lies in modifying the clustering subroutine which RCAST depends upon. There exists a vast array of EM alternatives the present literature which may be better suited to the problem of abstraction synthesis. For instance, EM is reliant upon Gaussian likelihood comparisons and is thus biased toward ellipsoid clusters; it may instead be advantageous to employ a hierarchical clustering algorithm that is better equipped to generate irregularly shaped clusters with a focus on contiguity.

Acknowledgements

This work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-1637736, IIS-1651089, IIS-1724157), Intel, Raytheon, and Lockheed Martin. Peter Stone serves on the Board of Directors of Cogitai, Inc. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

[2] We note that T-UCT cannot model continuous state and so we use a simple tile coding over $3^{\dim(F)}$ uniform tiles that evenly divide the value space of each factor F.

References

Bradski, G. 2000. The opencv library. *Dr. Dobb's Journal of Software Tools*.

Fayyad, U., and Irani, K. 1993. Multi-interval discretization of continuous-valued attributes for classification learning.

Friedman, J. H.; Bentley, J. L.; and Finkel, R. A. 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)* 3(3):209–226.

Hengst, B. 2002. Discovering hierarchy in reinforcement learning with HEXQ. In *ICML*, volume 2, 243–250.

Jong, N. K., and Stone, P. 2005. State abstraction discovery from irrelevant state variables. In *IJCAI*, volume 8, 752–757.

Jonsson, A., and Barto, A. G. 2001. Automated state abstraction for options using the u-tree algorithm. *Advances in neural information processing systems* 1054–1060.

Jonsson, A., and Barto, A. 2005. A causal approach to hierarchical decomposition of factored MDPs. In *Proceedings of the 22nd international conference on Machine learning*, 401–408. ACM.

Jonsson, A., and Barto, A. 2006. Causal graph based decomposition of factored MDPs. *The Journal of Machine Learning Research* 7:2259–2301.

Jonsson, A., and Barto, A. 2007. Active learning of dynamic Bayesian networks in Markov decision processes. In *Abstraction, Reformulation, and Approximation*. Springer. 273–284.

Kocsis, L., and Szepesvári, C. 2006. Bandit based Monte-Carlo planning. In *Machine Learning: ECML 2006*. Springer. 282–293.

Kohavi, R. 1996. Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *KDD*, volume 96, 202–207. Citeseer.

Konidaris, G., and Barto, A. G. 2009. Skill discovery in continuous reinforcement learning domains using skill chaining. In *Advances in neural information processing systems*, 1015–1023.

Liu, B.; Xia, Y.; and Yu, P. S. 2000. Clustering through decision tree construction. In *Proceedings of the ninth international conference on Information and knowledge management*, 20–29. ACM.

McGovern, A., and Barto, A. G. 2001. Automatic discovery of subgoals in reinforcement learning using diverse density.

Menache, I.; Mannor, S.; and Shimkin, N. 2002. Q-cut: dynamic discovery of sub-goals in reinforcement learning. In *European Conference on Machine Learning*, 295–306. Springer.

Menashe, J., and Stone, P. 2015. Monte Carlo Hierarchical Model Learning. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 771–779. International Foundation for Autonomous Agents and Multiagent Systems.

Moore, A. W. 1994. The parti-game algorithm for variable resolution reinforcement learning in multidimensional state-spaces. In *Advances in neural information processing systems*, 711–718.

Quinlan, J. R., et al. 1992. Learning with continuous classes. In *5th Australian joint conference on artificial intelligence*, volume 92, 343–348. Singapore.

Reynolds, S. I. 2000. Adaptive resolution model-free reinforcement learning: Decision boundary partitioning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 783–790. Morgan Kaufmann Publishers Inc.

Stolle, M., and Precup, D. 2002. Learning options in reinforcement learning. In *International Symposium on Abstraction, Reformulation, and Approximation*, 212–223. Springer.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* 112(1):181–211.

Vigorito, C. M., and Barto, A. G. 2010. Intrinsically motivated hierarchical skill learning in structured environments. *IEEE Transactions on Autonomous Mental Development* 2(2):132–143.

Run, Skeleton, Run: Skeletal Model in a Physics-Based Simulation

Mikhail Pavlov,* Sergey Kolesnikov,* Sergey M. Plis*

*Reason8

Abstract

In this paper, we present an approach to solve a physics-based reinforcement learning challenge "Learning to Run" with objective to train physiologically-based human model to navigate a complex obstacle course as quickly as possible. The environment is computationally expensive, has a high-dimensional continuous action space and is stochastic. We benchmark state of the art policy-gradient methods and test several improvements, such as layer normalization, parameter noise, action and state reflecting, to stabilize training and improve its sample-efficiency. We found that the Deep Deterministic Policy Gradient method is the most efficient method for this environment and the improvements we have introduced help to stabilize training. Learned models are able to generalize to new physical scenarios, e.g. different obstacle courses.

1 Introduction

Reinforcement Learning (RL) (Sutton and Barto 1998) is a significant subfield of Machine Learning and Artificial Intelligence along with the supervised and unsupervised subfields with numerous applications ranging from trading to robotics and medicine. It has already achieved high levels of performance on Atari games (Mnih et al. 2015), board games (Silver et al. 2016) and 3D navigation tasks (Mnih et al. 2016; Jaderberg et al. 2016).

All of above tasks have one feature in common - there is always some well-defined reward function, for example, game score, which can be optimized to produce the required behaviour. Nevertheless, there are are many other tasks and environments, for which it is still unclear what is the "correct" reward function to optimize. And it is even a harder problem, when we talk about continuous control tasks, such as physics-based environments (Todorov, Erez, and Tassa 2012) and robotics (Gu et al. 2017).

Yet, recently a substantial interest is directed to research employing physics-based based environment. These environments are significantly more interesting, challenging and realistic than the well defined games; at the same time they are still simpler than real conditions with physical agents, while being cheap and more accessible. One of the interesting researches is the work of Schulman et al. where a simulated robot learned to run and get up off the ground (Schulman et al. 2015b). Another paper is by Heess et al. where the authors trained several simulated bodies on a diverse set of challenging terrains and obstacles, using a simple reward function based on forward progress (Heess et al. 2017).

To solve the problem of continuous control in simulation environments it has become generally accepted to adapt the reward signal for specific environment. Still it can lead to unexpected results when the reward function is modified even slightly, and for more advanced behaviors the appropriate reward function is often non-obvious. To address this problem, the community came up with several environment-independent approaches such as unsupervised auxiliary tasks (Jaderberg et al. 2016) and unsupervised exploration rewards (Pathak et al. 2017). All these suggestions are trying to solve the main challenge of reinforcement learning: how an agent can learn for itself, directly from a limited reward signal, to achieve best performance.

Besides the difficulty in defining the reward function, physically realistic environments usually have a lot of stochasticity, are computationally very expensive, and have high-dimensional action spaces. To support learning in such settings it is necessary to have a reliable, scalable and sample-efficient reinforcement learning algorithm. In this paper we evaluate several existing approaches and then improve upon the best performing approach for a physical simulator setting. We present the approach that we have used to solve the "Learning to run" – NIPS 2017 competition challenge[1] with an objective to learn to control a physiologically-based human model and make it run as quickly as possible. The model that we present here has won the third place at the challenge: https://www.crowdai.org/challenges/nips-2017-learning-to-run/leaderboards.

This paper proceeds as follows: first we review the basics of reinforcement learning, then we describe environment used in challenge and models used in our experiment, after that we present results of our experiments and finally we discuss the results and conclude the work.

[1] https://www.crowdai.org/challenges/nips-2017-learning-to-run

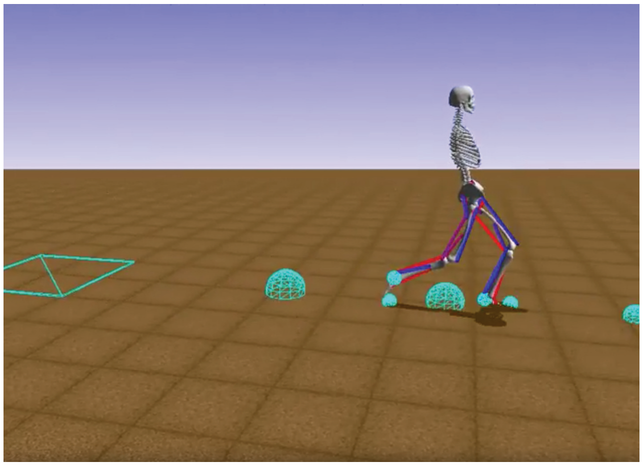


Figure 1: OpenSim screenshot that demonstrates the agent.

2 Background

We approach the problem in a basic RL setup of an agent interacting with an environment. The “Learning to run” environment is fully observable and thus can be modeled as a Markov Decision Process (MDP) (Bellman 1957). MDP is defined as a set of states ($\mathcal{S} : \{s_i\}$), a set of actions ($\mathcal{A} : \{a_i\}$), a distribution over initial states $p(s_0)$, a reward function $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$, transition probabilities $p(s_{t+1}|s_t, a_t)$, time horizon T, and a discount factor $\gamma \in [0, 1)$. A policy parametrized by θ is denoted with π_θ. The policy can be either deterministic, or stochastic. The agent’s goal is to maximize the expected discounted return $\eta(\pi_\theta) = \mathbb{E}_\tau[\sum_{t=0}^{T} \gamma_t r(s_t, a_t)]$, where $\tau = (s_0, a_0, \ldots, s_T)$ denotes a trajectory with $s_0 \sim p(s_0)$, $a_t \sim \pi_\theta(a_t|s_t)$, and $s_t \sim p(s_t|s_{t-1}, a_{t-1})$.

3 Environment

The environment is a musculoskeletal model that includes body segments for each leg, a pelvis segment, and a single segment to represent the upper half of the body (trunk, head, arms). See Figure 1 for a clarifying screenshot. The segments are connected with joints (e.g., knee and hip) and the motion of these joints is controlled by the excitation of muscles. The muscles in the model have complex paths (e.g., muscles can cross more than one joint and there are redundant muscles). The muscle actuators themselves are also highly nonlinear.

The purpose is to navigate a complex obstacle course as quickly as possible. The agent operates in a 2D world. The obstacles are balls randomly located along the agent’s way. Simulation is done using OpenSim (Delp et al. 2007) library which relies on the Simbody (Sherman, Seth, and Delp 2011) physics engine. The environment is described in Table 1. More detailed description of environment can be found on competition github page.[2]

Table 1: Description of the OpenSim environment.

parameters	description
state (s_t)	$\mathbb{R}^{41}$, coordinates and velocities of various body parts and obstacle locations. All (x, y) coordinates are absolute. To improve generalization of our controller and use data more efficiently, we modified the original version of environment making all x coordinates relative to the x coordinate of pelvis.
action (a_t)	$\mathbb{R}^{18}$, muscles activations, 9 per leg, each in $[0, 1]$ range.
reward	$\mathbb{R}$, change in x coordinate of pelvis plus a small penalty for using ligament forces.
terminal state	agent falls (pelvis $x < 0.65$) or 1000 steps in environment
stochasticity	• random strength of the psoas muscles • random location and size of obstacles

Due to a complex physics engine the environment is quite slow compared to standard locomotion environments (Todorov, Erez, and Tassa 2012; OpenAI Roboschool 2017). Some steps in environment could take seconds. Yet, the other environments can be as fast as three orders of magnitudes faster.[3] So it is crucial to train agent using the most sample-efficient method.

4 Methods

In this section we briefly describe the models we have evaluated in the task of the “Learning to run” challenge. We also describe our improvements to the model best performing in the competition: Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al. 2015).

4.1 On-policy methods

On-policy RL methods can only update agent’s behavior with data generated by the current policy. We consider two popular on-policy algorithms, namely Trust Region Policy Optimization (TRPO) (Schulman et al. 2015a) and Proximal Policy Optimization (PPO) (Schulman et al. 2017) as the baseline algorithms for environment solving.

Trust Region Policy Optimization (TRPO) is one of the notable state-of-the-art RL algorithms, developed by Schulman et al., that has theoretical monotonic improvement guarantee. As a basis, TRPO (Schulman et al. 2015a) using REINFORCE (Williams 1992) algorithm, that estimates the gradient of expected return $\nabla_\theta \eta(\pi_\theta)$ via likelihood ratio:

$$\nabla_\theta \eta(\pi_\theta) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t^i|s_t^i)(R_t^i - b_t^i), \quad (1)$$

[2] https://github.com/stanfordnmbl/osim-rl

[3] https://github.com/stanfordnmbl/osim-rl/issues/78

where N is the number of episodes, T is the number of steps per episode, $R_t^i = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}^i$ is the cumulative reward and b_t^t is a variance reducing baseline (Duan et al. 2016). After that, an ascent step is taken along the estimated gradient. TRPO improves upon REINFORCE by computing an ascent direction that ensures a small change in the policy distribution. As the baseline TRPO we have used the agent described in (Schulman et al. 2015a).

Proximal Policy Optimization (PPO) as TRPO tries to estimate an ascent direction of gradient of expected return that restricts the changes in policy to small values. We used clipped surrogate objective variant of proximal policy optimization (Schulman et al. 2017). This modification of PPO is trying to compute an update at each step that minimizes following cost function:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}_t[\min(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)], \tag{2}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta old}(a_t|s_t)}$ is a probability ratio (the new divided by the old policy), $\hat{A}_t = R_t - b_t$ is empirical return minus the baseline. This cost function is very easy to implement and allows multiple epochs of minibatch updates.

4.2 Off-policy methods

In contrast to on-policy algorithms, off-policy methods allow learning based on all data from arbitrary policies. It significantly increases sample-efficiency of such algorithms relative to on-policy based methods. Due to simulation speed litimations of the environment, we will only consider Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al. 2015).

Deep Deterministic Policy Gradient (DDPG) consists of actor and critic networks. Critic is trained using Bellman equation and off-policy data:

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma Q(s_{t+1}, \pi_\theta(s_{t+1})), \tag{3}$$

where π_θ is the actor policy. The actor is trained to maximize the critic's estimated Q-values by back-propagating through critic and actor networks. As in original article we used replay buffer and the target network to stabilize training and more efficiently use samples from environment.

DDPG improvements Here we present our improvements to the DDPG method. We used some standard reinforcement learning techniques: action repeat (the agent selects action every 5th state and selected action is repeated on skipped steps) and reward scaling. After several attempts, we choose a scale factor of 10 (i.e. multiply reward by ten) for our experiments. For exploration we used Ornstein-Uhlenbeck (OU) process (Uhlenbeck and Ornstein 1930) to generate temporally correlated noise for efficient exploration in physical environments. Our DDPG implementation was parallelized as follows: n processes collected samples with fixed weights all of which were processed by the learning process at the end of an episode, which updated their weights. Since DDPG is an off-policy method, the stale weights of the samples only improved the performance providing each sampling process with its own weights and thus improving exploration.

Parameter noise Another improvement is the recently proposed parameters noise (Plappert et al. 2017) that perturbs network weights encouraging state dependent exploration. We used parameter noise only for the actor network. Standard deviation σ for the Gaussian noise is chosen according to the original work (Plappert et al. 2017) so that measure d:

$$d(\pi, \widetilde{\pi}) = \sqrt{\left(\frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_s[(\pi(s)_i - \widetilde{\pi}(s)_i)^2]\right)}, \tag{4}$$

where $\widetilde{\pi}$ is the policy with noise, equals to σ in OU. For each training episode we switched between the action noise and the parameter noise choosing them with 0.7 and 0.3 probability respectively.

Layer norm Henderson et al. showed that layer normalization (Ba, Kiros, and Hinton 2016) stabilizes the learning process in a wide range of reward scaling. We have investigated this claim in our settings. Additionally, layer normalization allowed us to use same perturbation scale across all layers despite the use of parameters noise (Plappert et al. 2017). We normalized the output of each layer except the last for critic and actor by standardizing the activations of each sample. We also give each neuron its own adaptive bias and gain. We applied layer normalization before the nonlinearity.

Actions and states reflection symmetry The model has bilateral body symmetry. State components and actions can be reflected to increase sample size by factor of 2. We sampled transitions from replay memory, reflected states and actions and used original states and actions as well as reflected as batch in training step. This procedure improves stability of learned policy. If we dont use this step our model learned suboptimal policies, when for example muscles for only one leg are active and other leg just follows first leg.

5 Results

It this section we presents our experiments and setup. For all experiments we used environment with 3 obstacles and random strengths of the psoas muscles. We tested models on setups running 8 and 20 threads. For comparing different PPO, TRPO and DDPG settings we used 20 threads per model configuration. We have compared various combinations of improvements of DDPG in two identical settings that only differed in the number of threads used per configuration: 8 and 20. The goal was to determine whether the model rankings are consistent when the number of threads changes. For n threads (where n is either 8 or 20) we used $n-2$ threads for sampling transitions, 1 thread for training, and 1 thread for testing. For all models we used identical architecture of actor and critic networks. All hyperparameters are listed in Table 2. Our code used for competition

Table 2: Hyperparameters used in the experiments.

parameters	Value
Actor network architecture	$[64, 64]$, elu activation
Critic network architecture	$[64, 32]$, tanh activation
Actor learning rate	linear decay from $1\mathrm{e}{-3}$ to $5\mathrm{e}{-5}$ in $10\mathrm{e}6$ steps with Adam optimizer
Critic learning rate	linear decay from $2\mathrm{e}{-3}$ to $5\mathrm{e}{-5}$ in $10\mathrm{e}6$ steps with Adam optimizer
Batch size	200
γ	0.9
replay buffer size	5e6
rewards scaling	10
parameter noise probability	0.3
OU exploration parameters	$\theta = 0.1$, $\mu = 0$, $\sigma = 0.2$, $\sigma_{min} = 0.05$, $dt = 1\mathrm{e}{-2}$, n_{steps} annealing $\sigma_{\text{decay}}1\mathrm{e}6$ per thread

and described experiments can be found in a github repo.[4] Experimental evaluation is based on the undiscounted return $\mathbb{E}_\tau[\sum_{t=0}^{T} r(s_t, a_t)]$.

5.1 Benchmarking different models

Comparison of our winning model with the baseline approaches is presented in Figure 2. Among all methods the DDPG significantly outperformed PPO and TRPO. The environment is time expensive and method should utilized experience as effectively as possible. DDPG due to experience replay (re)uses each sample from environment many times making it the most effective method for this environment.

5.2 Testing improvements of DDPG

To evaluate each component we used an ablation study as it was done in the rainbow article (Hessel et al. 2017). In each ablation, we removed one component from the full combination. Results of experiments are presented in Figure 3a and Figure 3b for 8 and 20 threads respectively. The figures demonstrate that each modification leads to a statistically significant performance increase. The model containing all modifications scores the highest reward. Note, the substantially lower reward in the case, when parameter noise was employed without the layer norm. One of the reasons is our use of the same perturbation scale across all layers, which does not work that well without normalization. Also note, the behavior is quite stable across number of threads, as well as the model ranking. As expected, increasing the number of threads improves the result.

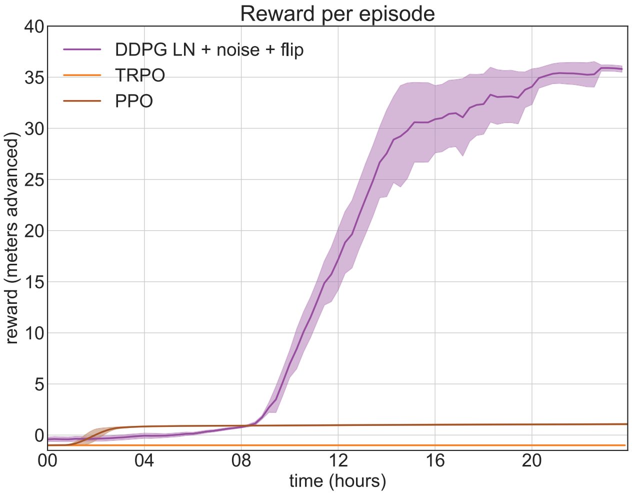


Figure 2: Comparing test reward of the baseline models and the best performing model that we have used in the "Learning to run" competition.

Table 3: Best achieved reward for each DDPG modification.

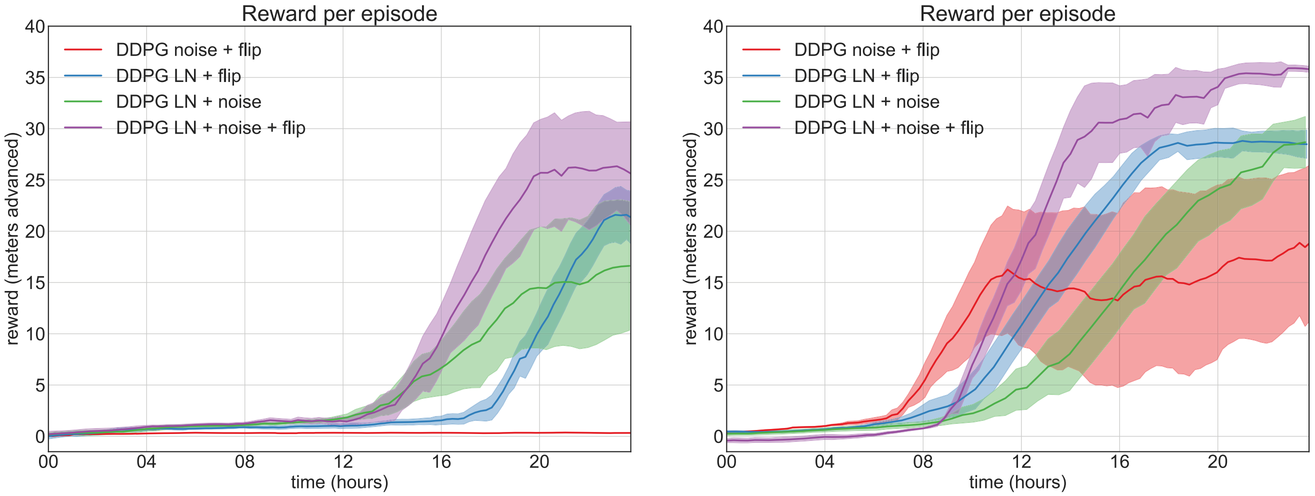
# threads / agent	8	20
DDPG + noise + flip	0.39	23.58
DDPG + LN + flip	25.29	31.91
DDPG + LN + noise	25.57	30.90
DDPG + LN + noise + flip	**31.25**	**38.46**

Maximal rewards achieved in the given time for 8 and 20 threads cases for each of the combinations of the modifications is summarized in Table 3. The main things to observe is a substantial improvement effect of the number of threads, and stability in the best and worst model rankings, although the models in the middle are ready to trade places.

6 Conclusions

Our results in OpenSim experiments indicate that in a computationally expensive stochastic environments that have high-dimensional continuous action space the best performing method is off-policy DDPG. We have tested 3 modifications to DDPG and each turned out to be important for learning. Action states reflection doubles the size of the training data and improves stability of learning and encourages the agent to learn to use left and right muscles equally well. With this approach the agent truly learns to run. Examples of the learned policies with and without the reflection are present at this URL https://tinyurl.com/ycvfq8cv. Parameter and Layer noise additionally improves stability of learning due to introduction of state dependent exploration. In general, we believe that investigation of human-based agents in physically realistic environments is a promising direction for future research.

[4] Theano: https://github.com/fgvbrt/nips_rl and PyTorch: https://github.com/Scitator/Run-Skeleton-Run



(a) 8 threads

(b) 20 threads

Figure 3: Comparing test reward for various modifications of the DDPG algorithm with 8 threads per configuration (Figure 3a) and 20 threads per configuration (Figure 3b). Although the number of threads significantly affects performance, the model ranking approximately stays the same.

References

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bellman, R. 1957. A markovian decision process. *Journal of Mathematics and Mechanics* 679–684.

Delp, S. L.; Anderson, F. C.; Arnold, A. S.; Loan, P.; Habib, A.; John, C. T.; Guendelman, E.; and Thelen, D. G. 2007. Opensim: open-source software to create and analyze dynamic simulations of movement. *IEEE transactions on biomedical engineering* 54(11):1940–1950.

Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; and Abbeel, P. 2016. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, 1329–1338.

Gu, S.; Holly, E.; Lillicrap, T.; and Levine, S. 2017. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, 3389–3396. IEEE.

Heess, N.; Sriram, S.; Lemmon, J.; Merel, J.; Wayne, G.; Tassa, Y.; Erez, T.; Wang, Z.; Eslami, A.; Riedmiller, M.; et al. 2017. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*.

Henderson, P.; Islam, R.; Bachman, P.; Pineau, J.; Precup, D.; and Meger, D. 2017. Deep reinforcement learning that matters. *arXiv preprint arXiv:1709.06560*.

Hessel, M.; Modayil, J.; Van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M.; and Silver, D. 2017. Rainbow: Combining improvements in deep reinforcement learning. *arXiv preprint arXiv:1710.02298*.

Jaderberg, M.; Mnih, V.; Czarnecki, W. M.; Schaul, T.; Leibo, J. Z.; Silver, D.; and Kavukcuoglu, K. 2016. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Mnih, V.; Badia, A. P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; and Kavukcuoglu, K. 2016. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, 1928–1937.

OpenAI Roboschool. 2017. OpenAI Roboschool (https://github.com/openai/roboschool).

Pathak, D.; Agrawal, P.; Efros, A. A.; and Darrell, T. 2017. Curiosity-driven exploration by self-supervised prediction. *arXiv preprint arXiv:1705.05363*.

Plappert, M.; Houthooft, R.; Dhariwal, P.; Sidor, S.; Chen, R. Y.; Chen, X.; Asfour, T.; Abbeel, P.; and Andrychowicz, M. 2017. Parameter space noise for exploration. *arXiv preprint arXiv:1706.01905*.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015a. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 1889–1897.

Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; and Abbeel, P. 2015b. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and

Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sherman, M. A.; Seth, A.; and Delp, S. L. 2011. Simbody: multibody dynamics for biomedical research. *Procedia Iutam* 2:241–261.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *Nature* 529(7587):484–489.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Todorov, E.; Erez, T.; and Tassa, Y. 2012. Mujoco: A physics engine for model-based control. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 5026–5033. IEEE.

Uhlenbeck, G. E., and Ornstein, L. S. 1930. On the theory of the brownian motion. *Physical review* 36(5):823.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Inverse Reinforcement Learning via Nonparametric Subgoal Modeling

Adrian Šošić
Signal Processing Group
Technische Universität Darmstadt
adrian.sosic@spg.tu-darmstadt.de

Abdelhak M. Zoubir
Signal Processing Group
Technische Universität Darmstadt
zoubir@spg.tu-darmstadt.de

Heinz Koeppl
Bioinspired Communication Systems
Technische Universität Darmstadt
heinz.koeppl@bcs.tu-darmstadt.de

Abstract

Recent advances in the field of inverse reinforcement learning (IRL) have yielded sophisticated frameworks which relax the original modeling assumption that the behavior of an observed agent reflects only a single intention. Instead, the demonstration data is separated into parts to account for the fact that different trajectories may correspond to different intentions, e.g., because they were generated by different domain experts. In this work, we go one step further: using the intuitive concept of *subgoals*, we build upon the premise that even a single trajectory can be explained more efficiently *locally* within a certain context than globally, enabling a more compact representation of the observed behavior. Based on this assumption, we build an implicit intentional model of the agent's goals to forecast its behavior in unobserved situations. The result is an integrated Bayesian prediction framework which provides spatially smooth policy estimates that are consistent with the expert's plan and significantly outperform existing IRL solutions. In addition, the framework can be naturally extended to handle scenarios with time-varying expert intentions.

1 Introduction

Inverse Reinforcement Learning (IRL) refers to the problem of inferring the intention of an agent, called *the expert*, from observed behavior. Under the Markov decision process (MDP) formalism (Sutton and Barto 1998), this intention is encoded in form of a reward function, which provides the agent an instantaneous feedback for each situation that can be encountered during the decision-making process. While early IRL methods such as (Ng and Russell 2000; Abbeel and Ng 2004; Ziebart et al. 2008; Ramachandran and Amir 2007; Levine, Popovic, and Koltun 2011) assume a single *global* reward function to explain the entire data set, recent methods relax this modeling assumption by allowing the intention of the agent to change with time (Nguyen, Low, and Jaillet 2015), or by hypothesizing that the data set is composed of several parts. For example, Dimitrakakis and Rothkopf (2011) propose a hierarchical prior over reward functions to account for the fact that different trajectories in the data set may reflect different intentions, e.g., because they were generated by different domain experts. Similarly, Babes et al. (2011) follow an expectation-maximization based clustering approach, grouping individual trajectories according to their underlying reward functions. Choi and Kim (2012) extended this idea by proposing a nonparametric Bayesian model in which the number of intentions is a priori unbounded.

In this work, we go a step further and start from the premise that, even in the case of a single expert or trajectory, the demonstrated behavior can be explained more efficiently *locally* within a certain context than by a single global reward model. As an illustrative example, consider the task shown in Fig. 2, where the expert approaches a set of intermediate goal positions before finally heading toward a global goal state. Despite the simplicity of the task, the encoding of such a behavior in a global intention model requires a reward structure that contains a comparably large number of redundant state-action based rewards. Alternative solution strategies include task-dependent expansions of the agent's state representation, e.g., to memorize the last visited goal (Krishnan et al. 2016), or they resort to more general decision-making frameworks like semi-MDPs / options (Bradtke and Duff 1995; Sutton, Precup, and Singh 1999) to achieve the necessary level of abstraction.

By contrast, the framework presented in this paper employs a rather simple, time-invariant representation of the task based on the intuitive concept of *subgoals*, which allows to efficiently encode the expert behavior using a task-appropriate partitioning of the state space. Our framework is based on the principle of Bayesian nonparametric inverse reinforcement learning (BNIRL) (Michini and How 2012) which, in its original form, is unable to generalize from the expert behavior. Building upon concepts from Bayesian policy recognition (Šošić, Zoubir, and Koeppl 2018), which applies a similar form of state space clustering but on a purely subintentional level, we remedy this deficit and extend the BNIRL idea to learn a compact implicit intentional model of the expert's goals. The result is an integrated Bayesian prediction framework which provides spatially smooth policy estimates that are consistent with the expert's plan and significantly outperform existing IRL solutions. Interestingly enough, our algorithm outperforms the baseline methods even when the expert's true reward structure is dense and the underlying subgoal assumption is violated.

2 Related Work: Revisiting BNIRL

The goal of Bayesian nonparametric inverse reinforcement learning is to build a model for the intention of an agent using demonstration data. Based on an MDP description of the observed task, the problem is formalized on a finite state space $\mathcal{S}$ using a time-invariant state transition model $T : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \rightarrow [0, 1]$, where $\mathcal{A}$ is a finite set of actions available to the agent at each state. For notational convenience, we represent the states in $\mathcal{S}$ by the integer values $\{1, \ldots, |\mathcal{S}|\}$, where $|\mathcal{S}|$ denotes the cardinality of $\mathcal{S}$. In BNIRL, it is assumed that we can observe a number of D expert demonstrations provided in the form of state-action pairs, $\mathcal{D} = \{(s_d, a_d)\}_{d=1}^{D}$, where each pair $(s_d, a_d) \in \mathcal{S} \times \mathcal{A}$ consists of a state s_d visited by the agent and the corresponding action a_d taken. Note that the model makes no assumption on the temporal ordering of the demonstrations, i.e., each pair is considered to arise from a specific but arbitrary time instant of the agent's decision-making process. In the following, we write $\mathbf{s} := \{s_d\}_{d=1}^{D}$ and $\mathbf{a} := \{a_d\}_{d=1}^{D}$ to access the collections of expert states and actions, respectively.

In contrast to the classical MDP formalism and most other IRL frameworks, BNIRL does *not* presuppose that the observed expert behavior necessarily originates from a single underlying reward function. Instead, it introduces the concept of *subgoals* (and corresponding *subgoal assignments*) with the underlying assumption that, at each decision instant, the expert selects a particular subgoal to plan the next action. Each subgoal is herein represented by a certain reward function defined on the system state space; in the simplest case, it corresponds to a single reward mass placed at a particular goal state in $\mathcal{S}$, which we identify with a reward function $R_g : \mathcal{S} \rightarrow \{0, C\}$ of the form

$$R_g(s) := \begin{cases} C & \text{if } g = s, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $g \in \{1, \ldots, |\mathcal{S}|\}$ indicates the respective subgoal location and $C \in (0, \infty)$ is some positive constant. Note that, in BNIRL, the number of subgoals is unbounded, even though the state space is assumed to be finite. We summarize this infinite collection of subgoals in the multiset $\mathcal{G} = \{g_k\}_{k=1}^{\infty} \in \times_{k=1}^{\infty} \mathcal{S}$ and adopt the assumption that $p(\mathcal{G}) = \prod_{k=1}^{\infty} p_g(g_k)$. The subgoal assignment in BNIRL is implemented using a set of indicator variables $\tilde{\mathbf{z}} = \{\tilde{z}_d \in \mathbb{N}\}_{d=1}^{D}$ that annotate each demonstration pair (s_d, a_d) with its unique subgoal index. Having targeted a particular subgoal $g_{\tilde{z}_d}$ while being at some state s_d, the expert is assumed to choose the next action a_d according to a softmax decision rule, $\pi : \mathcal{A} \times \mathcal{S} \times \mathcal{S} \rightarrow [0, 1]$, which weighs the expected returns of all actions against one another,

$$\pi(a_d \,|\, s_d, g_{\tilde{z}_d}) \propto \exp\left\{\beta Q^*(s_d, a_d \,|\, g_{\tilde{z}_d})\right\}. \tag{2}$$

Herein, $Q^*(s, a \,|\, g)$ denotes the state-action value (or *Q-value*) (Sutton and Barto 1998) of action a at state s under an optimal policy for the subgoal reward function R_g.

The softmax policy π models the expert's ability to maximize the future expected return in view of the targeted subgoal, while the coefficient $\beta \in [0, \infty)$ is used to express the expert's level of confidence in the optimal action. The prior distribution over indicators $p(\tilde{\mathbf{z}})$ is modeled by a Chinese restaurant process (CRP) (Aldous 1985), which assigns the event that indicator $\tilde{z}_d$ points to the jth subgoal the prior probability

$$p(\tilde{z}_d = j \,|\, \tilde{\mathbf{z}}_{\backslash d}) \propto \begin{cases} n_j & \text{if } j \in \{1, \ldots, K\}, \\ \alpha & \text{if } j = K + 1. \end{cases}$$

Herein, $\tilde{\mathbf{z}}_{\backslash d} := \{\tilde{z}_d\} \setminus \tilde{z}_d$ is a shorthand notation for the collection of all indicator variables without $\tilde{z}_d$. Further, n_j denotes the number of assignments to the jth subgoal in $\tilde{\mathbf{z}}_{\backslash d}$, K is the number of distinct entries in $\tilde{\mathbf{z}}_{\backslash d}$, and $\alpha \in [0, \infty)$ is a parameter controlling the diversity of the assignments. The joint distribution of demonstrated actions $\mathbf{a}$, subgoals $\mathcal{G}$, and subgoal assignments $\tilde{\mathbf{z}}$ (Fig. 1a) is thus given as

$$p(\mathbf{a}, \tilde{\mathbf{z}}, \mathcal{G} \,|\, \mathbf{s}) = p(\tilde{\mathbf{z}}) \prod_{k=1}^{\infty} p_g(g_k) \prod_{d=1}^{D} \pi(a_d \,|\, s_d, g_{\tilde{z}_d}). \tag{3}$$

Posterior inference in the BNIRL model refers to the (approximate) computation of the distribution $p(\tilde{\mathbf{z}}, \mathcal{G} \,|\, \mathcal{D})$, which allows to identify potential subgoal locations and the corresponding subgoal assignments based on the available demonstration data. For further details, the reader is referred to the original paper (Michini and How 2012).

2.1 Limitations of the BNIRL Model

Subgoal-based inference is a well motivated approach to IRL that has shown promising results. Yet, the original BNIRL model proposed by Michini and How comes with a significant drawback: due to its particular modeling assumptions, the framework is restricted to pure subgoal extraction and does *not* inherently provide a meaningful mechanism to forecast the expert behavior based on the inferred subgoals. The reason lies in the design of the framework which, at its heart, treats the subgoal assignments $\tilde{\mathbf{z}}$ as *exchangeable random variables* (Aldous 1985). By implication, the induced partitioning model $p(\tilde{\mathbf{z}})$ is agnostic about the covariate information in the data set and the resulting behavioral model is unable to reasonably propagate the expert knowledge to new situations.

To illustrate the problem, let us investigate the predictive action distribution that arises from the original BNIRL formulation. Without loss of generality of our claim, we may assume that the method has perfectly inferred all subgoals $\mathcal{G}$ and corresponding subgoal assignments $\tilde{\mathbf{z}}$ from the demonstration set. Denoting by a^* the predicted action at some new state s^*, the model yields

$$p(a^* \,|\, s^*, \mathcal{D}, \tilde{\mathbf{z}}, \mathcal{G}) = \sum_{\tilde{z}^*} p(a^* \,|\, \tilde{z}^*, s^*, \mathcal{D}, \tilde{\mathbf{z}}, \mathcal{G}) \;\times\; \ldots \tag{4}$$

$$\ldots\; p(\tilde{z}^* \,|\, s^*, \mathcal{D}, \tilde{\mathbf{z}}, \mathcal{G}) \overset{(\star)}{=} \sum_{\tilde{z}^*} p(a^* \,|\, \tilde{z}^*, s^*, g_{\tilde{z}^*}) p(\tilde{z}^* \,|\, \tilde{\mathbf{z}}),$$

where $\tilde{z}^*$ is the latent subgoal indicator belonging to s^*. Herein, $p(a^* \,|\, \tilde{z}^*, s^*, g_{\tilde{z}^*})$ can be either the softmax decision rule $\pi(a^* \,|\, s^*, g_{\tilde{z}^*})$ from Eq. (2) or the optimal deterministic policy for subgoal $g_{\tilde{z}^*}$, depending on whether we aspire to describe the *noisy expert behavior* at s^* or want to determine the *optimal action* according to the inferred reward model. Note that the second equality in Eq. (4), indicated by $(\star)$, follows from the conditional independence properties implied by Eq. (3), which can be easily verified using d-separation (Koller and Friedman 2009) on the graphical model in Fig. 1a.

As Eq. (4) reveals, the predictive model is characterized by the posterior distribution $p(\tilde{z}^* \mid s^*, \mathcal{D}, \tilde{\mathbf{z}}, \mathcal{G})$ of the latent subgoal assignment $\tilde{z}^*$ of state s^*. The intuition being that, in order to generalize the expert's plan to a new situation, we need to take into account the gathered information about what would be a likely subgoal targeted by the expert at s^*. Now, the problem with BNIRL is that the latter distribution is modeled without consideration of the actual state s^* (or any other observed variable) and effectively reduces $(\star)$ to the CRP prior $p(\tilde{z}^* \mid \tilde{\mathbf{z}})$, which due to its intrinsic exchangeability property only considers the frequency of the readily inferred subgoal assignments $\tilde{\mathbf{z}}$. Clearly, a subgoal selection strategy that is solely based on relative subgoal frequencies is of limited use when it comes to predicting the expert behavior at new states: the resulting subgoal assignment mechanism will inevitably ignore the structural information of the demonstration set and consistently produce the same subgoal assignment probabilities at all states, irrespective of the actual situation of the agent. By contrast, a reasonable assignment mechanism should inherently take into account the context of the agent's current state s^* when deciding about the next action.

Note in particular that the selection strategy proposed in Eq. (19) of (Michini and How 2012) falsely claims to solve this problem because the alleged conditioning on the query state has no actual effect on the involved subgoal indicator, as shown by Eq. (4). The only way to remedy the problem without modifying the model is via an external post-processing scheme such as the waypoint method described in (Michini et al. 2015).

3 Distance-Dependent Bayesian Inverse Reinforcement Learning

In this section, we introduce a redesigned inference framework, which we by analogy to BNIRL refer to as *distance-dependent Bayesian nonparametric IRL* (ddBNIRL). We derive our model by making two important modifications to the original BNIRL framework that address the previously described shortcomings on the conceptual level. We begin with an intermediate model, which introduces a subtle yet important structural modification to the BNIRL framework. In a second step, we generalize that new model to account for the structure of the control problem itself, which finally allows us to extrapolate our predictions to unseen states. As part of this generalization, we present a new state space metric that arises naturally in the context of ddBNIRL. In contrast to BNIRL, the resulting framework can be used likewise for *subgoal extraction* and *action prediction*. Moreover, following a Bayesian methodology, the presented approach provides complete posterior information at all levels.

3.1 The Intermediate Model

As a first step toward generalization, we establish a link between the model partitioning structure and the underlying system state space. For this purpose, we replace the demonstration-based indicators $\tilde{\mathbf{z}} = \{\tilde{z}_d \in \mathbb{N}\}_{d=1}^{D}$ with a new set of variables $\mathbf{z} := \{z_i \in \mathbb{N}\}_{i=1}^{|\mathcal{S}|}$. Unlike $\tilde{\mathbf{z}}$, these variables do not operate directly on the data but are instead tied to the elements in $\mathcal{S}$. Although they formally represent a new type of variable, we can still imagine that they are generated through a CRP. Accordingly, as illustrated in Fig. 1b, the joint distribution in Eq. (3) changes to

$$p(\mathbf{a}, \mathbf{z}, \mathcal{G} \mid \mathbf{s}) = p(\mathbf{z}) \prod_{k=1}^{\infty} p_g(g_k) \prod_{d=1}^{D} \pi(a_d \mid s_d, g_{\mathbf{z}_{s_d}}).$$

3.2 The ddBNIRL Model

The intermediate model makes it possible to reason about the policy (or, more suggestively, the underlying *state*-to-action rule) at visited parts of the state space. Yet, it is unable to extrapolate the gathered information to unvisited states, as explained in detail in Section 2.1. This problem is solved by replacing the exchangeable prior distribution induced by the CRP with a non-exchangeable one, generated by the distance-dependent Chinese restaurant process (ddCRP) (Blei and Frazier 2011). In contrast to the CRP, which assigns states to partitions, the ddCRP assigns states to other states, based on their pairwise distances. These "to-state" assignments are described by a set of indicators $\mathbf{c} = \{c_i \in \mathcal{S}\}_{i=1}^{|\mathcal{S}|}$ with prior distribution $p(\mathbf{c}) = \prod_{i=1}^{|\mathcal{S}|} p(c_i)$,

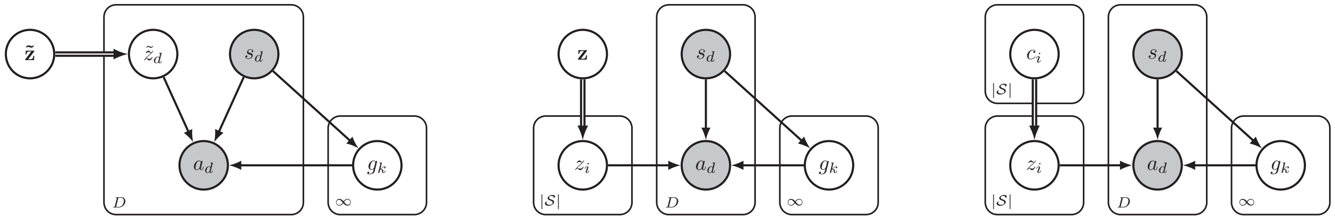
$$p(c_i = j) \propto \begin{cases} \nu & \text{if } i = j, \\ f(d_{i,j}) & \text{otherwise.} \end{cases}$$

Herein, $\nu \in [0, \infty)$ is called the self-link parameter of the process, $d_{i,j}$ denotes the distance from state i to state j, and $f : [0, \infty) \to [0, \infty)$ is a monotone decreasing score function. Note that the distances $\{d_{i,j}\}$ can be obtained via a suitable metric defined on the state space (see next paragraph). The state partitioning is then determined by the connected components of the induced ddCRP graph. Our ddBNIRL joint distribution (Fig. 1c) thus reads as

$$p(\mathbf{a}, \mathbf{c}, \mathcal{G} \mid \mathbf{s}) = p(\mathbf{c}) \prod_{k=1}^{\infty} p_g(g_k) \prod_{d=1}^{D} \pi(a_d \mid s_d, g_{\mathbf{z}(\mathbf{c})|_{s_d}}), \quad (5)$$

where $\mathbf{z}(\mathbf{c})|_s$ denotes the partition label of state s arising from the considered indicator set $\mathbf{c}$.

The Canonical State Metric for ddBNIRL The use of the ddCRP as a prior model for the state partitioning in Eq. (5) inevitably requires some notion of distance between any two states of the system in order to compute the involved function scores $\{f(d_{i,j})\}$. When no such distance measure is provided by the problem setting, a suitable (quasi-)metric can be derived from the transition dynamics of the system, which turns out to be the canonical choice for our subgoal problem. Consider the Markov chain governing the state process $\{s_t\}$ of an agent under some specific policy π. For



(a) BNIRL model

(b) intermediate model

(c) ddBNIRL model

Figure 1: Comparison of all discussed models. Shaded nodes: observed variables. Double strokes: deterministic dependencies.

any ordered pair of states (i, j), the chain naturally induces a value $\mathrm{T}^{\pi}_{i\to j}$, called a *hitting time* (Taylor and Karlin 2014; Tewari and Bartlett 2008), which represents the expected number of steps required until the state process, initialized at state i, eventually reaches state j for the first time,

$$\mathrm{T}^{\pi}_{i\to j} := \mathbb{E}\big[\min\{t \in \mathbb{N} : s_t = j\} \mid s_0 = i, \pi\big].$$

In the context of our subgoal problem, the natural quasimetric to measure the directed distance between two states i and j is thus given by the time it takes to reach the goal state j from the starting state i under the corresponding optimal subgoal policy $\pi_j(s) := \arg\max_{a\in\mathcal{A}} Q^*(s, a \mid j)\}$, i.e., $d_{i,j} := \mathrm{T}^{\pi_j}_{i\to j}$. For ddBNIRL, this choice is particularly appealing since the subgoal policies $\{\pi_j\}$ are already available during the inference procedure. The corresponding distances $\{d_{i,j}\}$ can be computed efficiently in a single policy evaluation step since they correspond to the optimal (negative) expected returns at the starting states for the special setting where the respective target state is made absorbing with zero reward while all other states are assigned a reward of -1. Note that, in order to implement a desired degree of locality in the model, the scale of the decay function f can be easily calibrated based on the quantiles of the resulting distances.

4 Prediction and Inference

Having introduced our subgoal model, we now explain how it can be used to generalize the expert behavior. We first focus on the task of action prediction at a given query state and then explain in a second step how the necessary information can be extracted from the demonstration data. Along the way, we also give insights into the implicit intentional model learned through the framework.

4.1 Action Prediction

Similar to the work by Abbeel and Ng (2004), we consider the task of predicting an action $a^* \in \mathcal{A}$ at some query state $s^* \in \mathcal{S}$ that is optimal with respect to the expert's *unknown* reward model. However, in contrast to most existing IRL methods, our approach is not based on point estimates of the expert's reward function but takes into account the entire hypothesis space of reward models to compute the full posterior predictive policy from the expert data. Mathematically, this task is formulated as computing the predictive action distribution $p(a^* \mid s^*, \mathcal{D})$, which captures the entire information about the expert behavior contained in the demonstration set $\mathcal{D}$. We start by expanding the distribution with the help of the latent state assignments $\mathbf{c}$,

$$p(a^* \mid s^*, \mathcal{D}) = \sum_{\mathbf{c}\in\mathcal{S}^{|\mathcal{S}|}} p(a^* \mid s^*, \mathcal{D}, \mathbf{c}) p(\mathbf{c} \mid \mathcal{D}).$$

The conditional distribution $p(a^* \mid s^*, \mathcal{D}, \mathbf{c})$ can be expressed in terms of the posterior distribution of the subgoal targeted at the query state s^*,

$$p(a^* \mid s^*, \mathcal{D}) = \sum_{\mathbf{c}\in\mathcal{S}^{|\mathcal{S}|}} p(\mathbf{c} \mid \mathcal{D}) \times \ldots$$

$$\ldots \sum_{i\in\mathcal{S}} p(a^* \mid s^*, \mathbf{c}, g_{\mathbf{z}(\mathbf{c})|_{s^*}} = i) p(g_{\mathbf{z}(\mathbf{c})|_{s^*}} = i \mid \mathcal{D}, \mathbf{c}),$$

where we used the fact that the prediction a^* is conditionally independent of the demonstration set $\mathcal{D}$ given the state partitioning structure and the corresponding subgoal assigned to s^* (that is, given $\mathbf{c}$ and $g_{\mathbf{z}(\mathbf{c})|_{s^*}}$). From the joint distribution in Eq. (5), it follows that

$$p(g_k \mid \mathcal{D}, \mathbf{c}) = \frac{1}{Z(\mathcal{D}, \mathbf{c})} p_g(g_k) \prod_{d:\mathbf{z}(\mathbf{c})|_{s_d} = k} \pi(a_d \mid s_d, g_k), \quad (6)$$

where $Z(\mathcal{D}, \mathbf{c})$ is the corresponding normalizing constant. Using this relationship, we get

$$p(a^* \mid s^*, \mathcal{D}) = \sum_{\mathbf{c}\in\mathcal{S}^{|\mathcal{S}|}} \frac{1}{Z(\mathcal{D}, \mathbf{c})} p(\mathbf{c} \mid \mathcal{D}) \sum_{i\in\text{supp}(p_g)} p_g(g_{\mathbf{z}(\mathbf{c})|_{s^*}} = i)$$

$$\ldots \times \prod_{d:\mathbf{z}(\mathbf{c})|_{s_d} = \mathbf{z}(\mathbf{c})|_{s^*}} \pi(a_d \mid s_d, g_{\mathbf{z}(\mathbf{c})|_{s^*}} = i) p(a^* \mid s^*, \mathbf{c}, g_{\mathbf{z}(\mathbf{c})|_{s^*}} = i).$$

In contrast to the summation over states, whose computational complexity is determined by the support of the subgoal prior distribution p_g and which grows at most linearly with the size of $\mathcal{S}$, the marginalization with respect to the indicator variables $\mathbf{c}$ involves the summation of $|\mathcal{S}|^{|\mathcal{S}|}$ terms and becomes quickly intractable even for small state spaces. Therefore, we approximate this operation via Monte Carlo integration, which yields

$$p(a^* \mid s^*, \mathcal{D}) \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{i\in\text{supp}(p_g)} p(g_{\mathbf{z}(\mathbf{c}^{\{n\}})|_{s^*}} = i \mid \mathcal{D}, \mathbf{c}^{\{n\}})$$

$$\ldots \times p(a^* \mid s^*, \mathbf{c}^{\{n\}}, g_{\mathbf{z}(\mathbf{c}^{\{n\}})|_{s^*}} = i),$$

where $\mathbf{c}^{\{n\}} \sim p(\mathbf{c} \mid \mathcal{D})$. The final prediction step can be performed, e.g., via the maximum a posteriori (MAP) policy estimate, $\hat{\pi}(s^*) := \arg\max_{a^*\in\mathcal{A}} p(a^* \mid s^*, \mathcal{D})$. Our inference task, hence, reduces to the computation of the posterior samples $\{\mathbf{c}^{\{n\}}\}$, which is described in the next section.

4.2 Partition Inference

Based on the joint model in Eq. (5), we obtain the posterior distribution $p(\mathbf{c}\,|\,\mathcal{D})$ in factorized form as

$$\begin{aligned} p(\mathbf{c}\,|\,\mathcal{D}) &= p(\mathbf{c}) \prod_{k=1}^{\infty} \sum_{g_k\in\text{supp}(p_g)} p(g_k) \prod_{d=1}^{D} \pi(a_d\,|\,s_d, g_{\mathbf{z}(\mathbf{c})|_{s_d}}) \\ &= p(\mathbf{c}) \prod_{k=1}^{|\mathbf{z}(\mathbf{c})|} \sum_{g_k\in\text{supp}(p_g)} \prod_{d:s_d\in\mathcal{C}_k} \pi(a_d\,|\,s_d, g_k)p(g_k), \qquad (7) \end{aligned}$$

where $\mathcal{C}_k$ denotes the kth state cluster induced by the assignment $\mathbf{c}$, i.e., $\mathcal{C}_k := \{s \in \mathcal{S} : \mathbf{z}(\mathbf{c})|_s = k\}$, and $|\mathbf{z}(\mathbf{c})|$ is the total number of clusters defined by $\mathbf{c}$. As explained by Blei and Frazier (2011), the indicator samples $\{\mathbf{c}^{\{n\}}\}$ can be efficiently generated using a fast-mixing Gibbs chain. Starting from a given ddCRP graph defined by the subset of indicators $\mathbf{c}_{\setminus i} := \{c_j\} \setminus c_i$, the insertion of an additional edge c_i will result in one of three possible outcomes: in the case of adding a self-loop to the ddCRP graph ($c_i = i$), the underlying partitioning structure stays unaffected. Setting $c_i \neq i$ either leaves the structure unchanged (if the target state is already in the same cluster as state i), or creates a new link between two clusters. In the latter case, the involved clusters are merged, which corresponds to a merging of the associated sums in Eq. (7). According to those three cases, the conditional distribution for the Gibbs procedure is given as

$$p(c_i = j\,|\,\mathbf{c}_{\setminus i}, \mathcal{D}) \;\propto\; \dots \qquad (8)$$
$$\begin{cases} \nu & \text{if } i = j, \\ f(d_{i,j}) & \text{if no clusters are merged,} \\ f(d_{i,j})\dfrac{\mathcal{L}(\mathcal{C}_{z_i}\cup\mathcal{C}_{z_j})}{\mathcal{L}(\mathcal{C}_{z_i})\cdot\mathcal{L}(\mathcal{C}_{z_j})} & \text{if clusters } \mathcal{C}_{z_i} \text{ and } \mathcal{C}_{z_j} \text{ are merged,} \end{cases}$$

where $\mathcal{L}(\mathcal{C})$ denotes the marginal action likelihood of all demonstrations accumulated in state cluster $\mathcal{C}$,

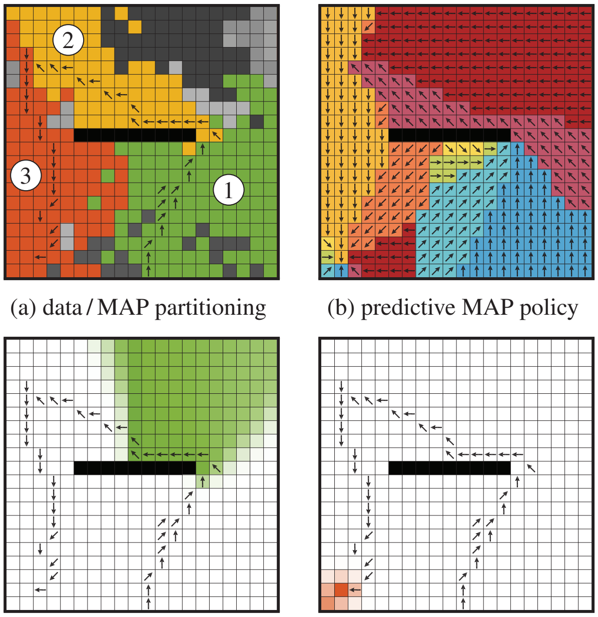
$$\mathcal{L}(\mathcal{C}) = \sum_{g\in\text{supp}(p_g)} \prod_{d:s_d\in\mathcal{C}} \pi(a_d\,|\,s_d, g)p_g(g).$$

4.3 Subgoal Inference

It is important to note that the sampling scheme described in the previous section is a collapsed one as all subgoals of our model are marginalized out during inference. In fact, the ddBNIRL framework differs from BNIRL and other IRL methods in that the reward model of the expert is never made explicit for predicting new actions. Nonetheless, if desired (e.g., for analyzing the expert's intentions), an estimate of the subgoal locations can be obtained in a post-hoc fashion from the subgoal posterior distribution in Eq. (6) for any given assignment $\mathbf{c}$. Two examples are provided in Fig. 2.

5 Simulation Results

In this section, we present experimental results for our algorithm, which we separate into two parts: a proof of concept and conceptual comparison to BNIRL, and a performance comparison with related algorithms.



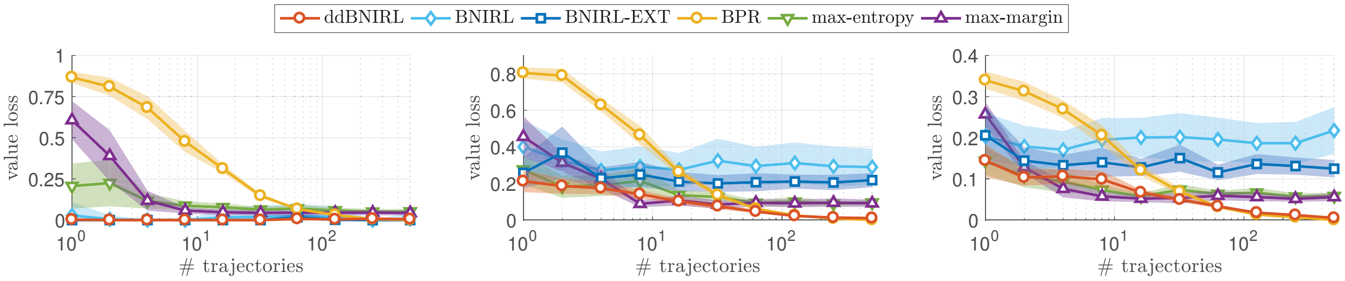
(a) data / MAP partitioning (b) predictive MAP policy

(c) first subgoal posterior (d) third subgoal posterior

Figure 2: Results for the original BNIRL data set.

5.1 Proof of Concept

To illustrate the differences to BNIRL and provide further insights into the latent intentional model learned through our framework, we adopt the original BNIRL data set with the state-action pairs depicted in Fig. 2a. The environment consists of $|\mathcal{S}| = 20 \times 20 = 400$ grid positions, where the black center bar indicates inaccessible wall states. Eight actions are available to the agent, which result in noisy state transitions toward the indicated motion directions. Figure 2 summarizes our results, which we computed from a posterior sample returned by our algorithm at a low temperature in a simulated annealing schedule (Kirkpatrick et al. 1983), assuming a uniform prior distribution over subgoals. Comparing the results to those in (Michini and How 2012), we observe three fundamental differences: i) in contrast to BNIRL, which has no built-in generalization mechanism, our method returns a predictive policy comprising the full posterior action information at all states (due to space constraints, we show only the MAP estimate); ii) exploiting the spatial context of the data, ddBNIRL is inherently robust to demonstration noise, resulting in a notably smoother data partitioning; iii) for each trajectory segment, we obtain an implicit representation of the associated subgoal in the form of a posterior distribution, without the need of assigning point estimates. Interestingly, the subgoal distribution corresponding to the green state partition has a comparably large spread on the upper side of the wall. This can be explained intuitively by the fact that any subgoal located in this high posterior region could have potentially caused the green state sequence, which circumvents the wall from the right. At the same time, the green area exhibits a sharp boundary on the left side since a subgoal located in the upper left region would have more likely resulted in a trajectory coming from the left.



(a) $N_R = 1$ ($\triangleq |\mathcal{S}|/100$)

(b) $N_R = 10$ ($\triangleq |\mathcal{S}|/10$)

(c) $N_R = 100$ ($\triangleq |\mathcal{S}|$)

Figure 3: Average value loss based on 100 Monte Carlo runs. The width of the shaded areas indicates one standard deviation.

5.2 Random MDP

To thoroughly evaluate the prediction accuracy of our model, we consider a class of randomly generated MDPs similar to the Garnet setting in (Bhatnagar et al. 2009). The transition dynamics $\{T(\cdot \mid s, a)\}$ are sampled independently from a symmetric Dirichlet distribution with concentration parameter 0.01, where we set $|\mathcal{S}| = 100$ and $|\mathcal{A}| = 10$. For each repetition of the experiment, N_R states are selected uniformly at random and assigned rewards that are, in turn, sampled uniformly from the interval $[0, 1]$. All other states are assigned zero reward. Then, we compute an optimal deterministic policy π^* with respect to a discount factor of 0.9 and generate a number of expert trajectories of length 10. Herein, the expert selects the optimal action with probability 0.9 and a random suboptimal action with probability 0.1. The obtained state sequences are passed to the algorithm and we compute the normalized value loss of the reconstructed policy as $\mathrm{L}(\pi^*, \hat{\pi}) := \|\mathbf{V}^* - \mathbf{V}^{\hat{\pi}}\|_2 \,/\, \|\mathbf{V}^*\|_2$, where $\mathbf{V}^*$ and $\mathbf{V}^{\hat{\pi}}$ represent the corresponding vectorized value functions.

As baseline methods, we adopt the subintentional Bayesian policy recognition (BPR) framework presented in (Šošić, Zoubir, and Koeppl 2018), as well as maximum-margin IRL (Abbeel and Ng 2004), maximum-entropy IRL (Ziebart et al. 2008), and vanilla BNIRL. Since BNIRL has no built-in generalization mechanism (Section 2.1) and the waypoint method does not straightforwardly apply to the considered scenario of multiple unaligned trajectories, we further compare our algorithm to an extension of BNIRL, which we refer to as BNIRL-EXT. Mimicking the ddBNIRL principle, the method accounts for the spatial context of the demonstrations by assigning each state to the BNIRL subgoal of the closest demonstration pair, according to the distance metric described in Section 3 — however, *after* the actual subgoal inference. When compared to ddBNIRL, this provides a reference of how much is effectively gained by modeling the spatial relationship of the data explicitly.

Figure 3 shows the loss over the size of the demonstration set for different reward settings. For small N_R, both BNIRL(-EXT) and ddBNIRL significantly outperform the remaining methods since the sparse reward structure allows for an efficient subgoal-based encoding of the expert behavior, which enables the algorithms to reconstruct the policy from even minimal amounts of demonstration data. However, the BNIRL(-EXT) solutions drastically deteriorate for denser reward structures. In particular, we observe a clear difference in performance between the cases where we do not account for the spatial information at all (BNIRL), include it in a post-processing step (BNIRL-EXT), and exploit it during the inference itself (ddBNIRL). Interestingly, ddBNIRL still outperforms the baselines in the dense reward regimes, even though the subgoal-based encoding loses its efficiency here. In fact, the results reveal that the proposed approach combines the merits of both model types, that is, the sample efficiency of the intentional models (max-margin / max-entropy) required for small data set sizes, as well as the asymptotic accuracy and fully probabilistic nature of the subintentional Bayesian framework (BPR).

6 Conclusion and Outlook

Based on BNIRL, we presented a novel method for data efficient IRL that leverages the spatial context of the demonstration set to learn a predictive model of the expert behavior. In the considered benchmark scenarios, it outperforms all reference methods while additionally capturing the full posterior information about the learned subgoal representation. However, like most other IRL algorithms, our method is agnostic about the *temporal context* of the demonstrations, and hence, the current model relies critically on the assumption that the expert's policy is *spatially consistent*. In fact, in its present form, the framework is unable to model trajectory crossings, because the diverging expert actions at the crossing points would need to be explained as a result of suboptimal behavior rather than as a consequence of changing intentions (e.g., when the expert spontaneously targets a different subgoal at an already visited state). For our algorithm to work, this means that either the underlying ground truth reward function has to be time-invariant or that it must be possible to resolve the expert's temporal subgoal schedule via an appropriate partitioning of the state space (as in Fig. 2). Currently, we extend the framework such that it can handle scenarios with time-varying goals. Based on the presented model, this requires only a minor step since the proposed distance-aware modeling concept allows us to incorporate the required temporal context by making only minor modifications, i.e., by switching from a spatial to an appropriate (spatio-)temporal distance metric. Preliminary results on real robotic data indicate a drastic reduction of the average subgoal localization error over BNIRL by more than 70%.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship Learning via Inverse Reinforcement Learning. In *International Conference on Machine Learning*, 1.

Aldous, D. J. 1985. *Exchangeability and Related Topics*. Springer.

Babes, M.; Marivate, V.; Subramanian, K.; and Littman, M. L. 2011. Apprenticeship Learning about Multiple Intentions. In *International Conference on Machine Learning*, 897–904.

Bhatnagar, S.; Sutton, R.; Ghavamzadeh, M.; and Lee, M. 2009. Natural Actor-Critic Algorithms. *Automatica* 45(11).

Blei, D. M., and Frazier, P. I. 2011. Distance Dependent Chinese Restaurant Processes. *Journal of Machine Learning Research* 12(Aug):2461–2488.

Bradtke, S. J., and Duff, M. O. 1995. Reinforcement Learning Methods for Continuous-Time Markov Decision Problems. In *Advances in Neural Information Processing Systems*, 393–400.

Choi, J., and Kim, K. 2012. Nonparametric Bayesian Inverse Reinforcement Learning for Multiple Reward Functions. In *Advances in Neural Information Processing Systems*, 305–313.

Dimitrakakis, C., and Rothkopf, C. A. 2011. Bayesian Multitask Inverse Reinforcement Learning. In *European Workshop on Reinforcement Learning*, 273–284.

Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P.; et al. 1983. Optimization by Simulated Annealing. *Science* 220(4598):671–680.

Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Krishnan, S.; Garg, A.; Liaw, R.; Miller, L.; Pokorny, F. T.; and Goldberg, K. 2016. HIRL: Hierarchical Inverse Reinforcement Learning for Long-Horizon Tasks with Delayed Rewards. `arXiv:1604.06508 [cs.RO]`.

Levine, S.; Popovic, Z.; and Koltun, V. 2011. Nonlinear Inverse Reinforcement Learning with Gaussian Processes. In *Advances in Neural Information Processing Systems*, 19–27.

Michini, B., and How, J. P. 2012. Bayesian Nonparametric Inverse Reinforcement Learning. In *Machine Learning and Knowledge Discovery in Databases*, 148–163.

Michini, B.; Walsh, T. J.; Agha-Mohammadi, A.-A.; and How, J. P. 2015. Bayesian Nonparametric Reward Learning from Demonstration. *IEEE Transactions on Robotics* 31(2):369–386.

Ng, A. Y., and Russell, S. J. 2000. Algorithms for Inverse Reinforcement Learning. In *International Conference on Machine Learning*, 663–670.

Nguyen, Q. P.; Low, B. K. H.; and Jaillet, P. 2015. Inverse Reinforcement Learning with Locally Consistent Reward Functions. In *Advances in Neural Information Processing Systems*, 1747–1755.

Ramachandran, D., and Amir, E. 2007. Bayesian Inverse Reinforcement Learning. *International Joint Conference on Artificial Intelligence* 1–4.

Šošić, A.; Zoubir, A. M.; and Koeppl, H. 2018. A Bayesian Approach to Policy Recognition and State Representation Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* `arXiv:1605.01278 [stat.ML]` Forthcoming.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

Sutton, R. S.; Precup, D.; and Singh, S. 1999. Between MDPs and Semi-MDPs: A Framework for Temporal Abstraction in Reinforcement Learning. *Artificial intelligence* 112(1-2):181–211.

Taylor, H. M., and Karlin, S. 2014. *An Introduction to Stochastic Modeling*. Academic Press.

Tewari, A., and Bartlett, P. L. 2008. Optimistic Linear Programming Gives Logarithmic Regret for Irreducible MDPs. In *Advances in Neural Information Processing Systems*, 1505–1512.

Ziebart, B. D.; Maas, A. L.; Bagnell, J. A.; and Dey, A. K. 2008. Maximum Entropy Inverse Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*, 1433–1438.

Multiagent Soft Q-Learning

Ermo Wei, Drew Wicke, David Freelan, Sean Luke
Department of Computer Science, George Mason University, Fairfax, VA USA
ewei@cs.gmu.edu, dwicke@gmu.edu, dfreelan@gmu.edu, sean@cs.gmu.edu

Abstract

Policy gradient methods are often applied to reinforcement learning in continuous multiagent games. These methods perform local search in the joint-action space, and as we show, they are susceptable to a game-theoretic pathology known as *relative overgeneralization*. To resolve this issue, we propose Multiagent Soft Q-learning, which can be seen as the analogue of applying Q-learning to continuous controls. We compare our method to MADDPG, a state-of-the-art approach, and show that our method achieves better coordination in multiagent cooperative tasks, converging to better local optima in the joint action space.

Introduction

Multiagent reinforcement learning (or MARL) is a type of Reinforcement Learning (RL) involving two or more agents. The mechanism is similar to traditional reinforcement learning: the environment is some current *state* (which the agent can only sense through its observation), the agents each perform some *action* while in that state, the agents each receive some *reward*, the state transitions to some new state, and than the process repeats. However in MARL, both transitions from state to state and the rewards allotted are functions of the *joint action* of the agents while in that state. Each agent ultimately tries to learn a *policy* that maps its observation to the optimal action in that state: but these are individual actions, not joint actions, as ultimately an agent cannot dictate the other agents' actions.

Multiagent Learning has been investigated comprehensively in discrete action domains. Many methods have been proposed for *equilibrium learning* (Littman 1994; 2001; Hu and Wellman 2003; Greenwald, Hall, and Serrano 2003), where the agents are trying to learn policies that satisfy some equilibrium concept from Game Theory. Almost all the equilibrium learning methods that have been proposed are based on off-policy Q-learning. This is not surprising, as multiagent equilibrium learning is naturally off-policy, that is, the agents are trying to learn an equilibrium policy while exploring the environment by following another policy. However, this situation does not apply to continuous games, that is, games with continuous actions. When RL must be applied to continuous control, policy gradient methods are often taken into consideration. However, in the past, it does not combine with off-policy samples as easily as the tabular Q-Learning. For this reason, RL has not been able to achieve as good performance in continuous games as it has in discrete domains.

In this paper, we consider cooperative games, where the agents all have the same reward function. Cooperative MARL problems can be categorized based on how much information each agent knows. If we have a central controller to control the learning process of each agent, then we have centralized training with decentralized execution (Oliehoek, Spaan, and Vlassis 2008). If the agents are learning concurrently, and each agent is told what the other agent or agents did, then the problem is known as a *joint action learner* problem. If the agents are learning concurrently but are not told what the others did, then we have an *independent learner* problem.

When the information is limited for learners in cooperative games, as is the case with independent learners, a pathology called *relative overgeneralization* can arise (Wei and Luke 2016). Relative overgeneralization occurs when a suboptimal Nash Equilibrium in the joint space of actions is preferred over an optimal Nash Equilibrium because each agent's action in the suboptimal equilibrium is a better choice when matched with arbitrary actions from the collaborating agents. For instance, consider a continuous game in Figure 1. The axes i and j are the various actions that agents A_i and A_j may perform (we assume the agents are performing deterministic actions), and the axis rewards (i, j) is the joint reward received by the agents from a given joint action $\langle i, j \rangle$. Joint action M has a higher reward than joint action N. However, the average of all possible rewards for action i_M, of agent A_i is lower than the average of all possible rewards for action i_N. Thus, the agents tend to converge to N.

In this paper, we first analytically show how relative overgeneralization prevents policy gradient methods from achieving better coordination in cooperative continuous games. This is even true in centralized training if we are not using the information wisely. Then we tackle the relative overgeneralization problem in these games by introducing Multiagent Soft Q-Learning, a novel method based on Soft Q-Learning and deep energy-based policies (Haarnoja et al. 2017). Our method is similar to MADDPG (Lowe et al.

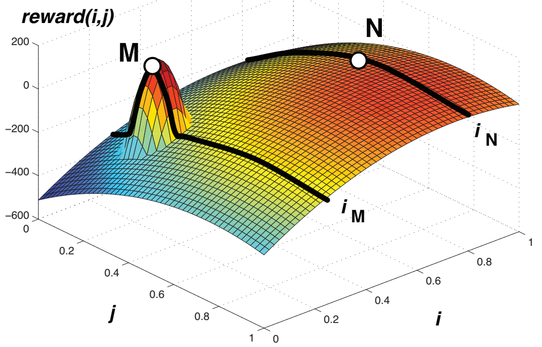


Figure 1: The relative overgeneralization pathology in continuous games.

2017), a recently proposed centralized learning algorithm. Thus, it belongs to the centralized training with decentralized execution paradigm. In this setting, since the training is centralized and we use the information wisely, it avoids the co-adaptation problem in Multiagent RL and greatly reduces the sample complexity, as the environment for agents is stationary.

Background

In this section, we first give an introduction to Markov Decision Processes (MDP) and various generalizations. We then introduce policy gradient methods.

Markov Decision Processes and Stochastic Games

A Markov Decision Process (or MDP) can be used to model the interaction an agent has with its environment. An MDP is a tuple $\{S, A, T, R, \gamma, H\}$ where S is the set of states; A is the set of actions available to the agent; T is the transition function $T(s, a, s') = P(s'|s, a)$ defining the probability of transitioning to state $s' \in S$ when in state $s \in S$ and taking action $a \in A$; R is the reward function $R : S \times A \mapsto \mathbb{R}$; $0 < \gamma < 1$ is a discount factor; and H is the horizon time of the MDP, that is, the number of steps the MDP runs.[1] An agent selects its actions based on the policy $\pi_\theta(a|s)$, which is a distribution over all possible actions a in state s parameterized by $\theta \in \mathbb{R}^n$.

The concept of an MDP can be extended to partially observable (POMDP) settings, where agents do not directly sense the state s. Rather, they receive some observation o sampled from a distribution conditioned on s.

MDPs can also be generalized to a cooperative multiagent settings, called a *Cooperative Stochastic Game* or CSG. This is a game with n agents (or players), defined by the tuple $\{S, \mathcal{A}, R, T, \gamma, H\}$, where S is the state space, $\mathcal{A} = A^1 \times ... \times A^n$ is the joint action space of n agents, $R : S \times \mathcal{A} \to \mathbb{R}$ is the reward function for each agent i, and $T(s, \vec{a}, s') = P(s'|s, \vec{a})$ is the transition function, where $\vec{a} = \langle a^1 \cdots a^n \rangle \in \mathcal{A}$ is the joint action of all agents. Thus the reward the agents receive and the state to which they transition depends on the current state and agents' joint action. Each agent i determines its action using a policy π^i. We will also use $-i$ to denote all agents except for agent i. A POCSG can be thought as taking CSG into a partially observable setting.

In the multiagent setting, a rational agent will play its *best response* to the other agents' strategy. If all agents are following a policy that implements this strategy, they will arrive at a Nash equilibrium defined as a solution where $\forall i \;\; R_i(s, \pi_1^*, \ldots \pi_i^* \ldots \pi_n^*) \geq R_i(s, \pi_1^*, \ldots, \pi_{i-1}^*, \pi_i, \pi_{i+1}^*, \ldots, \pi_n^*)$ for all of the strategies π_i available to agent i. π_i^* denotes the best response policy of agent i.

Policy Gradient Methods

In single agent continuous control tasks, it is common to apply a *policy gradient* method to determine an optimal policy. We describe that process here. To start, we define the expected return $J(\theta)$ of a policy π_θ as

$$J(\theta) = E_{P_\theta(\tau)}\big[R(\tau)\big] = \sum_\tau P_\theta(\tau) R(\tau), \qquad (1)$$

where $P_\theta(\tau)$ is the probability distribution over all possible state-action trajectories $\tau = \langle s_0, a_0, s_1, a_1, \ldots, s_H, a_H \rangle$ induced by following policy π_θ, and $R(\tau) = \sum_{t=0}^{H} \gamma^t R(s_t, a_t)$ is the discounted accumulated reward along trajectory τ. We want to compute the gradient $\nabla_\theta J(\theta)$, so that we can follow the gradient to a local optimum in the space of policy parameters. To do this we use the likelihood-ratio trick (Williams 1992), where we write the gradient as

$$\begin{aligned}\nabla_\theta J(\theta) =& \sum_\tau \nabla_\theta P_\theta(\tau) R(\tau) = \sum_\tau P_\theta(\tau) \frac{\nabla_\theta P_\theta(\tau)}{P_\theta(\tau)} R(\tau) \\ =& E_{P_\theta(\tau)}\big[\nabla_\theta \ln P_\theta(\tau) R(\tau)\big]\end{aligned} \qquad (2)$$

and estimate it by performing m sample trajectories $\langle \tau^{(1)}, \ldots, \tau^{(m)} \rangle$, calculating the corresponding terms, and then taking the average, that is, $\nabla_\theta J(\theta) \approx \frac{1}{m} \sum_{j=1}^{m} \nabla_\theta \ln P_\theta(\tau^{(j)}) R(\tau^{(j)})$. This policy gradient method can also use off-policy samples by introducing importance sampling, where we scale each term in the empirical expectation by $\frac{P_\theta(\tau)}{Q(\tau)}$, where Q is another distribution from which our off-policy samples come. The intuition behind Equation 2 is that the reward term $R(\tau)$ scales the gradient proportionally to the reward along the trajectories.

One problem with using this likelihood-ratio estimator in practice is that it suffers from a large variance, and thus requires a great many samples to give an accurate estimation. There are various methods proposed to deal with this. A first approach is to replace the Monte Carlo estimation

[1] Any infinite horizon MDP with discounted rewards can be ϵ-approximated by a finite horizon MDP using a horizon $H_\epsilon = \frac{\log_\gamma(\epsilon(1-\gamma))}{\max_{s,a}|R(s,a)|}$ (Jie and Abbeel 2010)

of the reward along trajectories $R(\tau)$ with a value function. This leads to the *Stochastic* (SPG) and *Deterministic Policy Gradient* (DPG) Theorems (Sutton et al. 1999; Silver et al. 2014), shown below respectively:

$$\begin{aligned}\nabla_\theta J(\theta) &= \int_S \rho^{\pi_\theta}(s) \int_A \nabla_\theta \pi_\theta(a|s) Q^{\pi_\theta}(s,a)\, da\, ds \\ &= E_{s\sim\rho^{\pi_\theta}, a\sim\pi_\theta}\Big[\nabla_\theta \ln \pi_\theta(a|s) Q^{\pi_\theta}(s,a)\Big] \\ \nabla_\theta J(\theta) &= \int_S \rho^{\pi_\theta}(s) \nabla_\theta \pi_\theta(s) \nabla_a Q^{\pi_\theta}(s,a)|_{a=\pi_\theta(s)}\, ds \\ &= E_{s\sim\rho^{\pi_\theta}}\Big[\nabla_\theta \pi_\theta(s) \nabla_a Q^{\pi_\theta}(s,a)|_{a=\pi_\theta(s)}\Big],\end{aligned}$$

where $\rho^{\pi_\theta}(s') = \int_S \sum_{t=1}^{\infty} \gamma^{t-1} P(s) P(s \to s', t, \pi_\theta)\, ds$ is the discounted distribution over states induced by policy π_θ and starting from some state $s \in S$. Specifically, $P(s \to s', t, \pi)$ is the probability of going t steps under policy π from state s and ending up in state s'. The theorems introduced a class of algorithms (Peters and Schaal 2008; Degris, White, and Sutton 2012) under the name *actor-critic* methods, where the *actor* is the policy π and the *critic* is the Q-function.

The actor-critic algorithms have also been used in an off-policy setting through importance sampling (Degris, White, and Sutton 2012). Recently, another method called a *replay buffer* (Mnih et al. 2013) has drawn people's attention for being able to do off-policy learning with actor-critic algorithms (Lillicrap et al. 2015). In this method, we store all the samples in a buffer and at every step of learning we sample a mini-batch from this buffer to estimate the gradient of either Q-Function or policy.

Related Work

The idea of learning to cooperate through policy gradient methods has been around for a long time, but mainly for discrete action domains (Banerjee and Peng 2003). Peshkin et al. have applied the REINFORCE policy gradient to both CSG and POCSG tasks. However, as we will show later, a naive use of this gradient estimator is dangerous in the multiagent case. Nair et al. proposed *Joint Equilibrium-Based Search for Policies* (JESP), applied to POCSGs. The main idea here is to perform policy search in one agent while fixing the policies of other agents. Although this method is guaranteed to converge to a local Nash Equilibrium, it is essentially a round-robin single agent algorithm.

Recently, with the boom of Deep Reinforcement Learning (DRL), deep MARL algorithms have been proposed to tackle large scale problems. One of the main streams is the centralized training with decentralized execution. Foerster et al. proposed a method to learn communication protocols between the agents. They use inter-agent backpropagation and parameter sharing. Foerster et al. studied how to stablize the training of multiagent deep reinforcment learning using importance sampling. Two actor-critic algorithms have been proposed in (Foerster et al. 2017a; Lowe et al. 2017). They argue that by using a central critic we can ease the training of multiple agents, and that by keeping a separate policy, the agent can execute with only its local information, which makes it possible to learn in POCSGs. Among these two algorithms, MADDPG (Lowe et al. 2017) is most relevant to us. It uses the learning rule from DDPG (Lillicrap et al. 2015) to learn a central off-policy critic based on Q-learning, and uses the following gradient estimator to learn the policies for each agent i:

$$\nabla_{\theta^i} J(\theta^i) = E_{s,a^{-i}\sim D}\Big[\nabla_{\theta^i} \pi^i_{\theta_i}(a_i|o_i) \nabla_{a_i} Q(s,\vec{a})|_{a_i=\pi^i_{\theta_i}(o_i)}\Big],$$

where θ^i is the agent i's policy parameters, D is the replay buffer, and o_i is the local observation of agent i. During the centralized training process, the critic has access to the true state $s = [o_1, \ldots, o_n]$. But at execution time, each agent only has access to o_i.

Multiagent Actor-Critic Algorithms

As we described earlier, if we have limited information for our agent, we can suffer from the relative overgeneralization problem. In this section, we demonstrate how this affects the actor critic algorithm. We will first derive the policy gradient estimator for the cooperative multiagent case and then discuss several problems that occurs if we naively use this estimator, which will shed light on the reasons why Multiagent Soft Q-Learning may be useful.

Proposition 1. *For any episodic cooperative stochastic game with n agents, we have the following multiagent stochastic policy gradient theorem:*

$$\begin{aligned}\nabla_{\theta^i} J(\vec{\theta}) = \int_s \rho^{\pi^1,\cdots,\pi^n}(s) \int_{A^i} \nabla_\theta \pi(a^i|s) \\ \int_{A^{-i}} \pi^{-i}(a^{-i}|s) Q^{\pi^1,\cdots,\pi^n}(s,\vec{a})\, da^{-i}\, da^i\, ds\end{aligned}$$

The proof of this proposition is provided in Proof A at the end of this paper. From Proposition 1, we can see that the policy gradient for agent i at each state is scaled by $Q^{\pi^i}(s,a^i) = \int_{A_{-i}} \pi^{-i}(a^{-i}|s) Q^{\pi^1\cdots\pi^n}(s,\vec{a}) da^{-i}$, which are the joint-action Q-values averaged by the other agents' policies. Their are several problems with this estimator. First, for any agent i, the joint-action Q-function is an on-policy Q-function. That is, it is learned under policy π^i, and π^{-i} which is not the best response policy of other agents. Thus, the joint-action Q-function may not scale the gradient in right magnitude. Second, if we are an independent learner and play as A_i in game shown in Figure 1, we only have access to $Q^{\pi^i}(s,a^i)$, since this value is averaged by other's policies, the value of the action $\langle i_N \rangle$ would be higher than the value of action $\langle i_M \rangle$ even under the optimal Q-function, thus mistakenly scaling the gradient to towards $\langle i_N \rangle$.

MADDPG solves the previous two issues by using the following methods. First, it uses the replay buffer (Lillicrap et al. 2015) to learn an off-policy optimal Q-function very much like what we learn for Q-Learning (Silver et al. 2014). This is not doable with traditional importance sampling based off-policy learning. Second, it's using the centralized training method which gives it direct access to the joint-action Q-function, but not the policies.

However, MADDPG fails to use the optimal action for gradient scaling, making it still vulnerable to the relative overgeneralization problem. To see that, consider its gradient estimator,

$$\nabla_{\theta^i} J(\theta^i) = E_{s,a^{-i}\sim D}\left[\nabla_{\theta^i}\pi^i_{\theta_i}(a_i|o_i)\nabla_{a_i}Q^*(s,\vec{a})|_{a_i=\pi^i_{\theta_i}(o_i)}\right]$$

We see that for agent i, it tries to ascend the policy gradient based on $Q^*(s,\vec{a})$, where a^{-i} is from the replay buffer D rather than the optimal policy, which is another way of averaging the Q-values based on others' policies. As we showed in Figure 1, this average-based estimation can lead to relative overgeneralization.

Multiagent Soft Q-Learning

In this paper, we propose MARL method for cooperative continuous games. We show that on the one hand, our method is an actor-critic method, which thus can benefit from the centralized training method, with one central critic and multiple distributed policies. And on the other hand, our method resembles Q-Learning, and thus, it efficiently avoids the relative overgeneralization problem. We first introduce Soft Q-Learning and then describe how we use it for multiagent training.

Soft Q-Learning

Although Q-Learning has been widely used to deal with control tasks, it has many drawbacks. One of the problems is that at the early stage of learning, the $\max$ operator can bring bias into the Q-value estimation (Fox, Pakman, and Tishby 2016). To remedy this, Maximum Entropy Reinforcment Learning (MERL) was introduced, in which tries to find the following policy:

$$\pi^*_{\text{MaxEnt}} = \text{argmax}_\pi \sum_t E_{(s_t,a_t)\sim\rho^\pi}[r(s_t,a_t) + \alpha\mathcal{H}(\pi(\cdot|s)]$$

where $\mathcal{H}(\pi(\cdot|s))$ is the entropy of the policy. The parameter α controls the relative importance of the reward and entropy: when it goes to 0, we recover ordinary RL. From this objective, a learning method similar to Q-Learning can be derived, called Soft Q-Learning (Haarnoja et al. 2017). Its learning algorithm is

$$Q_{\text{soft}}(s_t,a_t) \leftarrow r_t + \gamma E_{s_{t+1}}[V_{\text{soft}}(s_{t+1})] \quad \forall s_t, a_t,$$
$$V_{\text{soft}}(s_t) \leftarrow \alpha \log \int_A \exp(\frac{1}{\alpha}Q_{\text{soft}}(s_t,a'))da'.$$

Haarnoja et al. have shown that by using this update rule, Q_{soft} and V_{soft} can converge to Q^*_{soft} and V^*_{soft} respectively, and by driving $\alpha \to 0$, Q-learning with a hard max operator can be recovered. For this reason, Haarnoja et al. named this Soft Q-learning.

Once we have the learned Q-function above, we can get the optimal max entropy policy as

$$\pi^*_{\text{MaxEnt}}(a_t|s_t) = \exp(\frac{1}{\alpha}Q^*_{\text{soft}}(s_t,a_t) - V^*_{\text{soft}}(s_t)) \propto Q^*_{\text{soft}}(s_t,a_t).$$

A nice property of this policy is that it spreads widely over the entire action space in continuous control tasks. Thus, if we have such a policy, and if there are multiple modes in the action space, we can find them much more effectively than with more deterministic policies (e.g. Gaussian policy) which are typically used in actor-critic algorithms. However, since the form of this policy is so general, sampling from it is very hard. Soft Q-Learning solves this issue by using Stein Variational Gradient Descent (SVGD) (Liu and Wang 2016) to approximate the optimal policy through minimizing the KL-divergence:

$$D_{\text{KL}} = \left(\pi_\theta(\cdot|s_t)\middle|\middle| \exp\big(\frac{1}{\alpha}Q^*_{\text{soft}}(s_t,a_t) - V^*_{\text{soft}}(s_t)\big)\right), \quad (3)$$

where policy $\pi_\theta(\cdot|s)$ is our approximate policy. Since $-\frac{1}{\alpha}Q^*_{\text{soft}}(s_t,a_t)$ can be viewed as an energy function, and the authors are using a deep neural network to approximate the Q-function, they call this a *deep energy-based* policy. It has been demonstrated that using the Soft Q-Learning with deep energy based policies can learn multimodal objectives. In Soft Q-Learning we need to learn both the Q-function and the energy-based policy $\pi(\cdot|s)$. Thus, Soft Q-Learning can be thought as an actor-critic algorithm. Now consider the multiagent case. To make it clear, we first recall how we can achieve coordination in a discrete domain. In discrete domains, when we have the $Q^*(s,a)$ function, we simply apply the *argmax* operator to it and then let each agent do its own part of the optimal action. This is possible since we can do global search in the joint-action space for a given state. Now, with Soft Q-Learning and a deep energy-based model, we can mimic what we did in the discrete case. In this situation, we start with a high α to do global search in the joint-action space, then quickly anneal the α to lock on some optimal action, like the argmax operator. It has been shown that by annealing the α, we can get a deterministic policy from deep energy-based policies (Haarnoja et al. 2017).

Algorithm 1: Multiagent Soft Q-Learning

```
input: A central critic Q, N policies for all N agents,
       $\alpha$, and the epoch start to annealing $t$
for episode = 0 to M do
    Update central critic Q using the method from
      Soft Q-Learning.
    for agent = 1 to N do
        Update the joint policy for agent $i$ using
          equation (3)
    end
    if episode $\geq t$ then
        anneal $\alpha$
end
```

As we described before, Soft Q-Learning is also an actor-critic method. Thus, we can borrow the idea of learning a centralized joint action critic with Soft Q-Learning from MADDPG. Then for each of the agents, instead of learning a mapping for its own observation to its own action, we learn a mapping from its own observation to the joint-action. When the agent interacts with the environment, it just performs its own part of the joint action. We start the learning with high α value and let it explore the joint action space, we then quickly anneal the α to let each agent find a better local optima in joint-action space. Our algorithm is given at Algorithm 1.

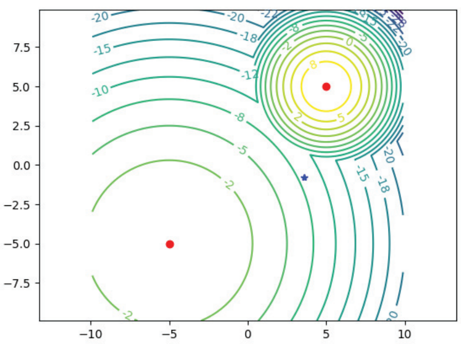


Figure 2: The Max of Two Quadratic game. The dots mark the two local optima in the joint action space while the star marks the joint action of the two agents. The contour shows the reward level.

Experiments

To show that our Multaigent Soft Q-Learning method can achieve better coordination, we consider the Max of Two Quadractics game from previous literature (Panait, Luke, and Wiegand 2006). This is a simple single state continuous game for 2 agents, one action dimension per agent. Each agent has a bounded action space. The reward for a joint action is given by following equation

$$f_1 = h_1 \times \big[-\big(\frac{a_1 - x_1}{s_1}\big)^2 - \big(\frac{a_2 - y_1}{s_1}\big)^2\big]$$
$$f_2 = h_2 \times \big[-\big(\frac{a_1 - x_2}{s_2}\big)^2 - \big(\frac{a_2 - y_2}{s_2}\big)^2\big)\big] + c$$
$$r(a_1, a_2) = \max(f_1, f_2)$$

where a_1, a_2 are the actions from agent 1 and agent 2 respectively. In the equation above, $h_1 = 0.8, h_2 = 1, s_1 = 3, s_2 = 1, x_1 = 5, x_2 = 5, y_1 = -5, y_2 = -5, c = 10$ are the coefficients to determine the reward surface (see Figure 2). Although the formulation of the game is rather simple, it poses a great difficulty to gradient-based algorithms as, over almost all the joint-action space, the gradient points to the sub-optimal solution located at (-5, -5).

We trained the MADDPG agent along with our Multagent Soft Q-Learning agent in this domain. As this was a simple domain, we used two-hidden-layer networks with size $\{100, 100\}$, and we trained the agents for 150 epochs for 100 steps per epoch. The training was not started until we had 1000 samples in the replay buffer. Both agents scaled their reward by 0.1. For our Multiagent Soft Q-Learning agent, we started the annealing at epoch 100, and finished the annealing in 15 epochs. We started with $\alpha = 1$, and annealed it to 0.001. For the rest of the parameters, we used the default setting from the original DDPG (Lillicrap et al. 2015) and Soft Q-Learning (Haarnoja et al. 2017) papers. In addition, to mimic the local observation setting where centralized learning was suitable, we gave the two agents in both algorithms different observation signals, where the first agent would always sense the state as $\langle 0 \rangle$, and the second agent would always sense it as $\langle 1 \rangle$. Then the state for the central critic was $\langle 0, 1 \rangle$.

The result is in Figure 3. However, the plot is an average over all 50 experiment runs, and hence, may hide some critical information. On closer investigation, we found that Multiagent Soft Q-Learning converged to the better equilibrium **72%** of the time, while MADDPG **never** converged to it.

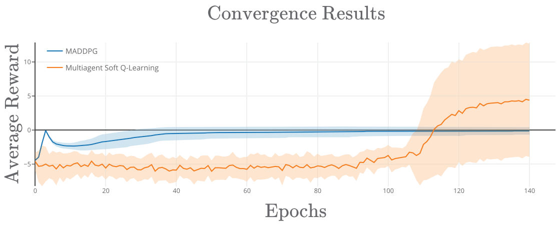


Figure 3: The average reward for both algorithms. Multiagent Soft Q-Learning finds the better local optima quickly after we anneal α.

Conclusion and Future Work

In this paper, we investigated how to achieve better controls in continuous games. We showed why the traditional policy gradient methods is not suitable for these tasks, and why the gradient-based method can fail to find better local optima in the joint-action space. We then proposed Multiagent Soft Q-Learning based on the centralized training and decentralized execution paradigm, and showed that, we can achieve much better coordination with higher probability. And since we are using centralized training, the co-adaption problem can be avoided, thus, making our method sample-efficient compared to independent learners. We argue that Multiagent Soft Q-Learning is a competitive RL learner for hard coordination problems.

There are some issues that we haven't been able to investigate thoroughly in this work. First, so far we have only applied our learner in the single state games. To better understand the algorithm, we would like to try our algorithm on sequential continuous games with hard coordination problems. Second, as we show in the experiment, with Soft Q-Learning we are not able to converge the better equilibrium for 100% of the time. In the future, we would like to investigate different annealing methods to improve the convergence rate. Last, we notice that Multiagent Soft Q-Learning models the joint action of all the agents, and thus the dimension of the action can explode with more agents. To solve this issue, we will investigate how to apply Soft Q-Learning in the independent learner case, where the algorithm scales well.

Acknowledgments

The research in this paper was conducted with the support of research infrastructure developed under NSF grant 1727303. We thank Tuomas Haarnoja and Haoran Tang for their helpful comments on implementing Soft Q-Learning.

Proof A

We first denote $\vec{\pi}$ as the joint-policy. This proof requires that $P(s), P(s'|s,\vec{a}), \pi^i(a^i|s), \nabla_{\theta^i}\pi^i(a^i|s)$, and $Q^{\vec{\pi}}(s,\vec{a})$ be continuous in all parameters and variables $s, s', \vec{a}$. This regularity condition implies that $V^{\vec{\pi}}(s)$ and $\nabla_{\theta^i}V^{\vec{\pi}}(s)$ are continuous functions of θ and s. S is also required to be compact, and so for any θ, $||\nabla_{\theta^i}V^{\vec{\pi}}(s)||$ is a bounded function of s. The proof mainly follows along the standard Stochastic Policy Gradient Theorem. We assume agent i follows the policy $\pi^i(a^i|s)$ parameterized by θ^i. We denote π^{-i} as the joint policy of all agents but agent i, and a^{-i} as the joint action of all agents except agent i. For notation simplicity, we denote:

$$\begin{aligned}
&\int_{A^{-i}} \pi^{-i}(a^{-i}|s) f(a^{-i})\, da^{-i} \\
=&\int_{A^1}\pi^1(a^1|s)\cdots\int_{A^{i-1}}\pi^{i-1}(a^{i-1}|s) \\
&\int_{A^{i+1}}\pi^{i+1}(a^{i+1}|s)\cdots\int_{A^n}\pi^n(a^n|s)f(a^{-i}) \\
&da^n\cdots\, da^{i+1}\, da^{i-1}\cdots\, da^1
\end{aligned}$$

Using this new notation the proof follows:

$$\begin{aligned}
&\nabla_{\theta^i}V^{\vec{\pi}}(s) \\
=&\nabla_{\theta^i}\int_{A^1}\pi^1(a^1|s)\cdots\int_{A^n}\pi^n(a^n|s)\, Q^{\vec{\pi}}(s,\vec{a})\, d\vec{a} \\
=&\int_{A^{-i}}\pi^{-i}(a^{-i}|s)\nabla_{\theta^i}\int_{A^i}\pi^i(a^i|s)Q^{\vec{\pi}}(s,\vec{a})\, da^i\, da^{-i} \\
=&\int_{A^{-i}}\pi^{-i}(a^{-i}|s)\int_{A^i}\Big[\nabla_{\theta^i}\pi^i(a^i|s)Q^{\vec{\pi}}(s,\vec{a}) \\
&\quad+\pi^i(a^i|s)\nabla_{\theta^i}Q^{\vec{\pi}}(s,\vec{a})\Big]\, da^i\, da^{-i} \\
=&\int_{A^{-i}}\pi^{-i}(a^{-i}|s)\int_{A^i}\Big[\nabla_{\theta^i}\pi^i(a^i|s)Q^{\vec{\pi}}(s,\vec{a}) \\
&\quad+\pi^i(a^i|s)\int_S\gamma P(s'|s,\vec{a})\nabla_{\theta^i}V^{\vec{\pi}}(s')\, ds'\Big]\, da^i\, da^{-i}
\end{aligned}$$

We used Leibniz integral rule to exchange order of derivative and integration using the regularity condition and expanding $Q^{\vec{\pi}}(s,\vec{a})$ above. We use $P^{\pi}(s'|s,t)$ as short for $P(s\to s',t,\pi)$. Now we iterate the formula,

$$\begin{aligned}
=&\int_{A^{-i}}\pi^{-i}(a^{-i}|s)\int_{A^i}\Big[\nabla_{\theta^i}\pi^i(a^i|s)Q^{\vec{\pi}}(s,\vec{a})+\pi^i(a^i|s) \\
&\int_S\gamma P(s'|s,\vec{a})\Big[\int_{A^{-i}}\pi^{-i}(a^{-i}|s')\int_{A^i}\Big(\nabla_{\theta^i}\pi^i(a^i|s')Q^{\vec{\pi}}(s',\vec{a}) \\
&+\pi^i(a^i|s')\int_S\gamma P(s''|s',\vec{a})\nabla_{\theta^i}V^{\vec{\pi}}(s'')\, ds''\Big)\, da^i\, da^{-i}\Big]\, ds'\Big] \\
&da^i\, da^{-i} \\
=&\int_{A^{-i}}\pi^{-i}(a^{-i}|s)\int_{A^i}\nabla_{\theta^i}\pi^i(a^i|s)Q^{\vec{\pi}}(s,\vec{a})da^i da^{-i} \\
&+\int_S\gamma P^{\vec{\pi}}(s'|s,1)\int_{A^{-i}}\pi^{-i}(a^{-i}|s') \\
&\int_{A^i}\nabla_{\theta^i}\pi^i(a^i|s')Q^{\vec{\pi}}(s',\vec{a})da^i da^{-i} ds' \\
&+\int_S\gamma P^{\vec{\pi}}(s'|s,1)\int_S\gamma P^{\vec{\pi}}(s''|s',1)\nabla_{\theta^i}V^{\vec{\pi}}(s'')\, ds''\, ds' \\
=&\int_S\sum_{t=0}^{\infty}\gamma^t P^{\vec{\pi}}(s'|s,t) \\
&\int_{A^{-i}}\pi^{-i}(a^{-i}|s')\int_{A^i}\nabla_{\theta^i}\pi^i(a^i|s')Q^{\vec{\pi}}(s',\vec{a})\, da^i\, da^{-i}\, ds' \\
=&\int_S\sum_{t=0}^{\infty}\gamma^t P^{\vec{\pi}}(s'|s,t) \\
&\int_{A^i}\nabla_{\theta^i}\pi^i(a^i|s')\int_{A^{-i}}\pi^{-i}(a^{-i}|s')Q^{\vec{\pi}}(s',\vec{a})\, da^{-i}\, da^i\, ds'
\end{aligned}$$

In the final line we use Fubini's theorem and exchange the order of integration using the regularity condition so that $||\nabla_{\theta^i}V^{\vec{\pi}}(s)||$ is bounded. We then take the expectation over the possible start states s:

$$\begin{aligned}
\nabla_{\theta^i}J(\theta)=&\nabla_{\theta^i}\int_S P(s)V^{\vec{\pi}}(s)\, ds \quad =\int_S P(s)\nabla_{\theta^i}V^{\vec{\pi}}(s)\, ds \\
=&\int_S\int_S\sum_{t=0}^{\infty}\gamma^t P(s)P^{\vec{\pi}}(s'|s,t)\, ds\int_{A_i}\nabla_{\theta^i}\pi^i(a^i|s') \\
&\int_{A_{-i}}\pi^{-i}(a^{-i}|s')Q^{\vec{\pi}}(s',\vec{a})\, da^{-i}\, da^i\, ds' \\
=&\int_S\rho^{\vec{\pi}}(s')\int_{A^i}\nabla_{\theta^i}\pi^i(a^i|s') \\
&\int_{A^{-i}}\pi^{-i}(a^{-i}|s')Q^{\vec{\pi}}(s',\vec{a})\, da^{-i}\, da^i\, ds' \qquad \square
\end{aligned}$$

References

Banerjee, B., and Peng, J. 2003. Adaptive policy gradient in multiagent learning. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, 686–692. New York, NY, USA: ACM.

Degris, T.; White, M.; and Sutton, R. S. 2012. Linear off-policy actor-critic. In *In International Conference on Machine Learning*. Citeseer.

Foerster, J.; Assael, Y. M.; de Freitas, N.; and Whiteson, S. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2137–2145.

Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2017a. Counterfactual multi-agent policy gradients. *arXiv preprint arXiv:1705.08926*.

Foerster, J.; Nardelli, N.; Farquhar, G.; Afouras, T.; Torr, P. H. S.; Kohli, P.; and Whiteson, S. 2017b. Stabilising experience replay for deep multi-agent reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, 1146–1155.

Fox, R.; Pakman, A.; and Tishby, N. 2016. Taming the noise in reinforcement learning via soft updates. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence*, UAI'16, 202–211.

Greenwald, A.; Hall, K.; and Serrano, R. 2003. Correlated Q-learning. In *AAAI Spring Symposium*, volume 3, 242–249.

Haarnoja, T.; Tang, H.; Abbeel, P.; and Levine, S. 2017. Reinforcement learning with deep energy-based policies. *ICML*.

Hu, J., and Wellman, M. P. 2003. Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research* 4:1039–1069.

Jie, T., and Abbeel, P. 2010. On a connection between importance sampling and the likelihood ratio policy gradient. In *Advances in Neural Information Processing Systems*, 1000–1008.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Littman, M. L. 1994. Markov games as a framework for multi-agent reinforcement learning. In *ICML*, volume 94, 157–163.

Littman, M. L. 2001. Friend-or-foe Q-learning in general-sum games. In *ICML*, volume 1, 322–328.

Liu, Q., and Wang, D. 2016. Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, 2378–2386.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems (NIPS)*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; and Riedmiller, M. 2013. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*.

Nair, R.; Tambe, M.; Yokoo, M.; Pynadath, D.; and Marsella, S. 2003. Taming decentralized pomdps: Towards efficient policy computation for multiagent settings. In *IJCAI*, 705–711.

Oliehoek, F. A.; Spaan, M. T.; and Vlassis, N. 2008. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research* 32:289–353.

Panait, L.; Luke, S.; and Wiegand, R. P. 2006. Biasing coevolutionary search for optimal multiagent behaviors. *IEEE Transactions on Evolutionary Computation* 10(6):629–645.

Peshkin, L.; Kim, K.-E.; Meuleau, N.; and Kaelbling, L. P. 2000. Learning to cooperate via policy search. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, 489–496. Morgan Kaufmann Publishers Inc.

Peters, J., and Schaal, S. 2008. Natural actor-critic. *Neurocomputing* 71(7):1180–1190.

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms. In *ICML*.

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; Mansour, Y.; et al. 1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*.

Wei, E., and Luke, S. 2016. Lenient learning in independent-learner stochastic cooperative games. *Journal of Machine Learning Research* 17(84):1–42.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Hierarchical Approaches for Reinforcement Learning in Parameterized Action Space

Ermo Wei, Drew Wicke, Sean Luke
Department of Computer Science, George Mason University, Fairfax, VA USA
ewei@cs.gmu.edu, dwicke@gmu.edu, sean@cs.gmu.edu

Abstract

We explore Deep Reinforcement Learning in a parameterized action space. Specifically, we investigate how to achieve sample-efficient end-to-end training in these tasks. We propose a new compact architecture for the tasks where the parameter policy is conditioned on the output of the discrete action policy. We also propose two new methods based on the state-of-the-art algorithms Trust Region Policy Optimization (TRPO) and Stochastic Value Gradient (SVG) to train such an architecture. We demonstrate that these methods outperform the state of the art method, Parameterized Action DDPG, on test domains.

Introduction

Deep Reinforcement Learning (DRL) has achieved success in recent years, including beating human masters in Go (Silver et al. 2016), attaining human level performance in Atari games (Mnih et al. 2015), and controlling robots in high-dimensional action spaces (Lillicrap et al. 2015). With these successes, researchers have begun to explore new frontiers in DRL, including how to apply DRL in complex action spaces. Consider for example the real time strategy game StarCraft, where at any time during play we may choose among different types of actions to be able to finish our goals (Vinyals et al. 2017). For example, we may need to choose a building to construct and then select where to build it; or choose a squad of armies and direct them to explore the map. Critically, instead of having a single action set, we may have several sets of actions, either continuous or discrete, and to get a meaningful action to execute, we must choose wisely among these sets.

In this paper, we explore how to apply DRL to tasks with more than one set of actions. Specifically, we consider tasks with parameterized action spaces (Masson, Ranchod, and Konidaris 2016), where at each step the agent must choose both a discrete action and a set of continuous parameters for that action. Tasks with this kind of action space have been proposed in the Reinforcement Learning (RL) community for a long time (Stone et al. 2006) but have not been explored much.

One approach to handle a RL task with a parameterized action space is to do *alternating optimization* (Masson, Ranchod, and Konidaris 2016). Here, we break the task into two separate subtasks by fixing either the parameters or discrete actions and then applying RL algorithms alternating between the induced subtasks. Although this method can work, it has a huge sample complexity because every time we switch the subtask, the previous experience is no longer valid as the environment is different.

Thus, a sample efficient alternative is to train the policies for discrete actions and parameters at the same time. There have already been steps in this direction. Hausknecht and Stone simultaneously train two policies which can produce the values for discrete action and parameters respectively and then select the action to execute based on their values. There are two main drawbacks of this method. The first is that the parameter policy does not know what discrete action is selected at execution time. Thus, the parameter policy needs to output all the parameters for all the discrete actions at every step. As a result, the output size of the parameter policy can explode if we have high dimensional parameters with large discrete action sets. The second problem is that neither the policies nor the training method are aware of the action-selection procedure after the action and parameter values are produced. Therefore, the method may be missing a crucial piece of information for it to succeed.

In this paper, we propose a new architecture for parameterized action space tasks. In our method, we condition our parameter policy on the output of the discrete action policy, thus greatly reducing the output size of the parameter policy. Then we extend the state-of-the-art DRL algorithms to efficiently train the new architecture for parameterized action space tasks. In experiments we show that our methods can achieve better performance than the state of the art method.

Background

Before we delve into the architecture and algorithms, we first present a mathematical formulation of Markov Decision Processes (MDPs) along with some relevant policy gradient algorithms. Then we present Parameterized Action MDPs (PAMDPs). Lastly, we discuss some of the previous work in PAMDPs that is related to our paper.

MDPs and Policy Gradient Methods

Markov Decision Process A Markov Decision Process (or MDP) can be used to model the interaction an agent has with its environment. A MDP is a tuple $\{S, A, T, R, \gamma, H\}$ where S is the set of states; A is the set of actions available to the agent; T is the transition function $T(s, a, s') = P(s'|s, a)$ defining the probability of transitioning to state $s' \in S$ when in state $s \in S$ and taking action $a \in A$; R is the reward function $R : S \times A \mapsto \mathbb{R}$; $0 < \gamma < 1$ is a discount factor; and H is the horizon time of the MDP, that is, the MDP runs for only H steps.* An agent selects its actions based on a policy $\pi_\theta(\cdot|s)$, which is a distribution over all possible actions a in state s parameterized by $\theta \in \mathbb{R}^n$.

Policy Gradient Methods One of the major approaches to deal with continuous control problems in MDPs is to apply a policy gradient method. In policy gradient methods, we are trying to use gradient ascent to optimize the following objective

$$\begin{aligned} J(\theta) &= E_{s\sim\rho^{\pi_\theta}}[V^{\pi_\theta}(s)] \\ &= \int_S \rho^{\pi_\theta}(s) V^{\pi_\theta}(s)\, ds, \end{aligned} \tag{1}$$

where s is the state visited, and $\rho^{\pi_\theta}(s)$ is the distribution over all states induced by executing policy π_θ. Many algorithms have been proposed to optimize this objective, including REINFORCE (Williams 1992), GPOMDP (Baxter and Bartlett 2001), and Trust Region Policy Optimization (TRPO) (Schulman et al. 2015), where we collect a set of trajectory samples with the form $\tau = \langle s_0, a_0, s_1, a_1, \ldots, s_H, a_H\rangle$ and use them to evaluate the gradient of $J(\theta)$. It turns out that sometimes, it is beneficial to learn an additional value function $Q(s, a)$ or $V(s)$ to reduce the variance in estimating the gradient of $J(\theta)$. This leads to a family of algorithms named "actor-critic" algorithms where the "actor" is the policy π and the "critic" is the value function. This family of algorithms includes the Stochastic Policy Gradient Theorem (SPG) (Sutton et al. 2000), the Deterministic Policy Gradient Theorem (DPG) (Silver et al. 2014), and so on. In addition, DDPG (Lillicrap et al. 2015) is an extention of DPG to the DRL setting by using a replay buffer to assist off-policy learning.

Parameterized Action MDPs

The MDP notation can be generalized to deal with parameterized tasks, e.g., actions with parameters. Here, instead of having just one set of actions, we have multiple sets of controls: a finite set of discrete actions $A_d = \{a_1, a_2, \ldots, a_n\}$ and for each $a \in A_d$, a set of continuous parameters $X_a \subseteq R^{m_a}$. Thus, an action is a tuple (a, x) in the joint action space,

$$A = \bigcup_{a\in A_d} \{(a, x)|x \in X_a\}.$$

MDPs with this action space are called Parameterized Action MDPs (PAMDPs) (Masson, Ranchod, and Konidaris 2016).

Previous Work on Parameterized Action MDPs

Tasks with parameterized actions have been a research topic in RL for a long time (Stone et al. 2006). Zamani et al. considered tasks with a set of discrete parameterized actions (2012). However, their algorithm which is based on Symbolic Dynamic Programming, is limited to MDPs with internal logical relations.

We adopt the Parameterized Action MDP setting from (Masson, Ranchod, and Konidaris 2016). In their work, they train the policy in an alternative fashion. They first fix all the parameter policies, and hence induce an MDP with action set A of only discrete actions. Then they use Q-learning to learn a discrete policy in that MDP, and upon convergence, they fix the discrete policy, and start training the parameter policy. They show that this method can converge to local optima.

Rachelson, Fabiani, and Garcia used parameterized actions to deal with continuous time MDPs where the parameter for all the actions is the waiting time (2009). Thus, they have a unified parameter space. Sharma, Lakshminarayanan, and Ravindran did a similar approach where they extended TRPO to control the repetition of the action, that is, how many steps an action should execute (2017). They argued that the repetition times can be considered as a parameter for their original control signal. However, the repetition times are drawn from a fix set of integers, which is not a continuous signal.

The method that has the closest connection to our work is (Hausknecht and Stone 2015), which extended the DDPG to a parameterized action space. In this algorithm, the policy outputs all the parameters and all the discrete actions, and then selects the (a, x) tuple with the highest Q-value.

Hierarchical Approaches in PAMDPs

In this paper we propose a new, more natural architecture to generate actions for parameterized action tasks. In our algorithm, we have one neural network for the discrete policy and one neural network for the parameter policy. Our parameter policy $\pi(x|s, a)$ takes two inputs, the state s and the discrete action a sampled from discrete action policy $\pi(a|s)$. Then the joint action is given by $(a, x) \sim \pi(a, x|s) = \pi(a|s)\pi(x|s, a)$. Since the action a is known before we generate the parameters, we do not need the post processing step of determining which action tuple (a, x) has the highest Q-values. And since the parameter policy knows the discrete action a, the output size of parameter policy remains constant.

Previously, this architecture was not plausible in policy gradient methods due to the fact that we have to sample the discrete action in the middle of the forward pass, and the gradient cannot flow back to the discrete action policy in the backward pass due to the sampling operation. In this section, we describe two algorithms, Parameterized Action TRPO (PATRPO) and Parameterized Action SVG(0) (PASVG(0)) that solve this problem.

Before we delve into the algorithms, we first introduce some notation. We use $\pi_\Theta(a, x|s)$ to denote our overall policy, where a is the discrete action, x is the the parame-

*Any infinite horizon MDP with discounted rewards can be ϵ-approximated by a finite horizon MDP using a horizon $H_\epsilon = \frac{\log_\gamma(\epsilon(1-\gamma))}{\max_{s,a}|R(s,a)|}$ (Jie and Abbeel 2010).

ter for that action, and Θ is all parameters for the model. Our policy can be broken into two separate policies using conditional probability $\pi_\Theta(a, x|s) = \pi^c_{\theta_x}(x|a, s)\pi^d_{\theta_a}(a|s)$, where θ_a and θ_x are the parameters for discrete action policy $\pi^d(a|s)$ and continuous parameter policy $\pi^c(x|a, s)$ respectively, and $\Theta = [\theta_a, \theta_x]$.

Parameterized Actions TRPO

Among all the policy gradient algorithms, TRPO and DDPG achieve the best performance as they are able to optimize large neural network policies (Duan et al. 2016). Thus, we consider how to apply these two algorithms in PAMDPs.

We first consider how to optimize our policy using TRPO's technique. In the TRPO, we are solving the following optimization problem:

$$\begin{aligned}&\text{maximize}_\theta \;\; L_{\theta'}(\theta) = E_{s\sim\rho_{\theta'}, a\sim\pi_{\theta'}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta'}(a|s)}Q_{\theta'}(s, a)\right]\\ &\text{subject to} \;\; \overline{\text{KL}}_{\theta'}(\theta) = E_{s\sim\rho_{\theta'}}[D_{\text{KL}}(\pi_{\theta'}(\cdot|s)||\pi_\theta(\cdot|s))] < \delta,\end{aligned}$$

where θ' and θ are the parameter vectors before and after each policy update respectively, and L is the surrogate loss. $Q_\theta(s, a)$ indicates the Q-function fitted using the samples from policy parameterized by θ. The idea behind TRPO is to optimize the policy in a stable way such that the new policy distribution after each update will not be too different from the old one. This is achieved through the KL-divergence constraint between the policy distributions before and after the parameter update.

A similar idea has been explored before in the natural policy gradient (Kakade 2002), where the objective function is replaced with linear approximation $\frac{\partial L_{\theta'}(\theta)}{\partial\theta}(\theta - \theta')$ and the KL-divergence is replaced with a quadratic approximation $(\theta' - \theta)^T A(\theta' - \theta)$. The positive semidefinite matrix A in the quadratic term is the Hessian matrix of constraint, e.g., $A = \frac{\partial^2}{\partial^2\theta}\overline{\text{KL}}_{\theta'}(\theta)$. However, when the policy model becomes large, A becomes very expensive to compute and store. What is special about TRPO is that it uses a Hessian-free optimization method (Martens 2010; Pearlmutter 1994) and conjugate gradient descent method to avoid the explicit formation of the Hessian matrix. Therefore, TRPO only has a slight increase in the computation cost for optimizing large neural networks.

To apply the TRPO in PAMDPs, we first write down the optimization problem using our notation, which is

$$\begin{aligned}&\text{maximize}_\Theta \;\; E_{s\sim\rho_{\Theta'}, (a,x)\sim\pi_{\Theta'}}\left[\frac{\pi_\Theta(a, x|s)}{\pi_{\Theta'}(a, x|s)}Q_{\Theta'}(s, a, x)\right]\\ &\text{subject to} \;\; E_{s\sim\rho_{\Theta'}}[D_{\text{KL}}(\pi_{\Theta'}(\cdot|s)||\pi_\Theta(\cdot|s))] < \delta\end{aligned}$$

The objective can be further expanded to

$$E_{s\sim\rho_{\Theta'}, (a,x)\sim\pi_{\Theta'}}\left[\frac{\pi^c_{\theta_x}(x|a, s)\pi^d_{\theta_a}(a|s)}{\pi^c_{\theta'_x}(x|a, s)\pi^d_{\theta'_a}(a|s)}Q_{\Theta'}(s, a, x)\right]$$

Notice that, in the objective function, the samples are collecting using Θ' instead of Θ. Thus, in training time, we can just take the gradient of objective function w.r.t Θ to achieve end-to-end training like normal supervised learning, and do not need to use any trick.

However, some changes are needed to meet the constraint of TRPO as there is no closed form solution for computing KL-divergence between two joint distributions. Here, we rewrite the KL-divergence constraint into conditional divergence using the chain rule.

$$\begin{aligned}&E_{s\sim\rho_{\Theta'}}[D_{\text{KL}}(\pi_{\Theta'}(\cdot|s)||\pi_\Theta(\cdot|s))]\\ =&E_{s\sim\rho_{\Theta'}}\Bigg[D_{\text{KL}}(\pi^d_{\theta'_a}(\cdot|s)||\pi^d_{\theta_a}(\cdot|s))\\ &+ E_{a\sim\pi^d_{\theta'_a}(a|s)}\left[D_{\text{KL}}(\pi^c_{\theta'_x}(\cdot|s, a)||\pi^c_{\theta_x}(\cdot|s, a))\right]\Bigg]\\ =&E_{s\sim\rho_{\Theta'}}\left[D_{\text{KL}}(\pi^d_{\theta'_a}(\cdot|s)||\pi^d_{\theta_a}(\cdot|s))\right]\\ &+ E_{s\sim\rho_{\Theta'}}E_{a\sim\pi^d_{\theta'_a}(a|s)}\left[D_{\text{KL}}(\pi^c_{\theta'_x}(\cdot|s, a)||\pi^c_{\theta_x}(\cdot|s, a))\right]\end{aligned}$$

Thus, we can use samples to estimate both the objective function and KL-divergence. However, we notice that we can further reduce the variance of estimating the KL-divergence by using the analytical form of discrete action policy $\pi(a|s)$. That is, the KL-divergence can be written as

$$\begin{aligned}&E_{s\sim\rho_{\Theta'}}\left[D_{\text{KL}}(\pi^d_{\theta'_a}(\cdot|s)||\pi^d_{\theta_a}(\cdot|s))\right]\\ &+ E_{s\sim\rho_{\Theta'}}\left[\pi(a|s)D_{\text{KL}}(\pi^c_{\theta'_x}(\cdot|s, a)||\pi^c_{\theta_x}(\cdot|s, a))\right]\end{aligned}$$

Using this form of constraint allows us to estimate the divergence between two policies even when we do not have samples for some discrete actions.

Parameterized Actions SVG(0)

Now we propose our second method based on the reparameterization trick.

One thing that makes the policy gradient methods special is that the samples we need to estimate the gradient come from the policy we are optimizing. That is, the objective usually takes the following form,

$$E_{p_\theta(x)}[f(x)].$$

We can write the gradient of expectation w.r.t θ in this way:

$$\begin{aligned}\frac{\partial E_{p_\theta(x)}[f(x)]}{\partial\theta} &= \frac{\partial}{\partial\theta}\int_x p_\theta(x)f(x)\,dx\\ &= \int_x \frac{\partial p_\theta(x)}{\partial\theta}f(x)\,dx.\end{aligned}$$

Since we lost the term $p(x)$ in the integral after we take the gradient, it's no longer an expectation, hence, we can no longer use samples from $p(x)$ to estimate it.

To solve this problem, people made the following changes to the gradient,

$$\begin{aligned}\frac{\partial E_{p_\theta(x)}[f(x)]}{\partial\theta} &= \int_x \frac{\partial p_\theta(x)}{\partial\theta}f(x)\,dx\\ &= \int_x p(x)\left(\frac{1}{p(x)}\frac{\partial p_\theta(x)}{\partial\theta}\right)f(x)\,dx\\ &= E_{p_\theta(x)}\left[\frac{\partial \ln p_\theta(x)}{\partial\theta}f(x)\right]\end{aligned}$$

This trick is the foundation for most of the policy gradient methods in RL.

Recently, another trick has been used to attack the same problem in the unsupervised learning community (Kingma and Welling 2013; Rezende, Mohamed, and Wierstra 2014). The idea is that a continuous random variable z can be obtained by first taking a noise variable ϵ and then deterministically transforming it. For example, a gaussian random variable $z \sim \mathcal{N}(\mu, \sigma^2)$ can be reparameterized into a noise random variable $\epsilon \sim \mathcal{N}(0, 1)$ with a deterministc transformation $g_{\mu,\sigma}(z) = \mu + \sigma\epsilon$. By applying this technique, we can optimize an expectation using samples from a noise distribution as follows

$$E_{p_\theta(x)}[f(x)] = \int_x p_\theta(x) f(x)\, dx = \int_\epsilon p(\epsilon) f(g_\theta(\epsilon))\, d\epsilon$$

Then the gradient can be easily written as

$$\frac{\partial E_{p_\theta(x)}[f(x)]}{\partial \theta} = \int_\epsilon p(\epsilon) \left(\frac{\partial f}{\partial g} \frac{\partial g}{\partial \theta} \right) d\epsilon = E_{p(\epsilon)} \left[\frac{\partial f}{\partial g} \frac{\partial g}{\partial \theta} \right]$$

This method has been successfully used in Variational Autoencoders (VAE) for various works (Walker et al. 2016; Sohn, Lee, and Yan 2015). It has also been applied to RL to train Stochastic Value Gradient (SVG) Learners (Heess et al. 2015). Recently, Jang, Gu, and Poole (2016), Maddison, Mnih, and Teh (2016) generalized the reparameterization trick to deal with discrete random variables with the *Gumbel-Softmax* trick. In the discrete case, a random variable x can be drawn from a discrete distribution $\{p(x_1), p(x_2), \ldots, p(x_n)\}$ by the Gumbel-Max trick (Maddison, Tarlow, and Minka 2014),

$$x = \mathrm{argmax}_i [g_i + \ln p(x_i)]$$

where $g_i \sim$ Gumbel$(0, 1)$. The Gumbel-Softmax trick replace the argmax operator in the above with a continuous differentiable softmax operator. With this change, we can now draw samples as

$$x = \frac{\exp\left[\left((g_i + \ln p(x_i)\right)/t\right]}{\sum_{i=1}^n \exp\left[\left((g_i + \ln p(x_i)\right)/t\right]}$$

where t is the “temperature” used to control the tradeoff between bias and variance. This trick has been applied to the RL setting as well, including imitation learning (Baram et al. 2017) and multiagent RL (Mordatch and Abbeel 2017).

For our problem, the key observation is that the two steps of decision making in a parameterized action policy (choosing from a discrete action and then determining the parameters for it), is very much like the paradigm in VAE (2013). In the VAE setting, the encoder of the VAE takes a sample x from the dataset, and generates a latent variable z from it. Then the decoder takes z and reconstructs x out of it. For our situation, the discrete action policy first takes the state s as input and generates a discrete action a, then determines the parameters x based on action a using the continuous parameter policy. Thus, we can roughly think of our discrete action policy and continuous parameter policy as the encoder and decoder in VAE respectively.

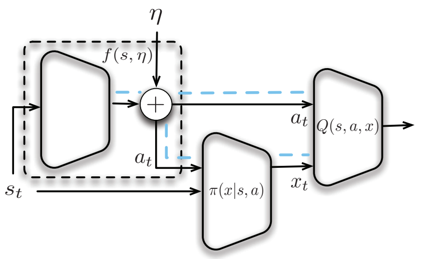


Figure 1: The training flow of the PASVG(0) agent. The black lines indicate the forward pass of the training, and the dash lines indicate the backward pass of the training. The dash box marks the reparameterized policy f.

We start with the objective function in (1) and write it in parameterized action setting.

$$\begin{aligned} J(\Theta) &= \int_s \rho^{\pi_\Theta}(s) V^{\pi_\Theta}(s) ds \\ &= E_{s \sim \rho_\Theta} \left[\sum_a \pi_\Theta(a, x|s) Q(s, a, x) \right] \\ &= E_{s \sim \rho_\Theta} \left[\sum_a \pi_{\theta_a}(a|s) Q(s, a, \pi_{\theta_x}(x|s, a)) \right] \end{aligned}$$

For the last step in the previous derivation, we use the DDPG formulation. Then we apply the reparameterization trick. Following the convention in (Heess et al. 2015), we use η to represent the auxiliary noise variable instead of ϵ in the VAE setting.

$$J(\Theta) = E_{s \sim \rho_\Theta} \left[\sum_\eta p(\eta) Q(s, f_{\theta_a}(s, \eta), \pi_{\theta_x}(s, f_{\theta_a}(s, \eta))) \right]$$

where $a = f_{\theta_a}(s, \eta)$ is the discrete action policy after reparameterization. Then the gradient w.r.t Θ is simply

$$\frac{\partial J(\Theta)}{\partial \Theta} = E_{\rho_\Theta} E_{p(\eta)} \left[\frac{\partial}{\partial \Theta} Q(s, f_{\theta_a}(s, \eta), \pi_{\theta_x}(s, f_{\theta_a}(s, \eta))) \right]$$

Since we are reparameterizing our stochastic policy for a 0-step value function (Q-function), similar to Heess et al.'s method, thus we name our algorithm Parameterized Action SVG(0) (See Figure 1 for the training flow).

However, there is one critical difference between our work and Heess et al.. In our work, we do not need to infer the noise variable since we are not using any dynamic model. To see this, we rewrite the gradient estimation using the Bayes' rule, following the method from (Heess et al. 2015),

$$\begin{aligned} \frac{\partial J(\Theta)}{\partial \Theta} = {} & E_{\rho_\Theta} E_{\pi(a,x|s)} E_{p(\eta|a,x,s)} \\ & \left[\frac{\partial}{\partial \Theta} Q(s, f_{\theta_a}(s, \eta), \pi_{\theta_x}(s, f_{\theta_a}(s, \eta))) \right] \qquad (2) \end{aligned}$$

Heess et al. use this method to infer the noise ζ of their reparameterized approximate dynamic model $s' =$

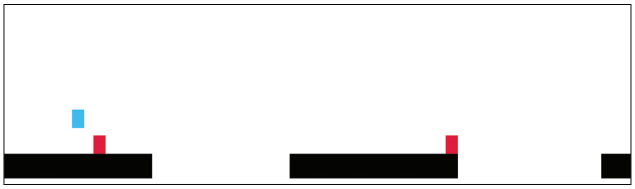


Figure 2: Platform domain

$g(s, a, \zeta)$. Thus, they need to learn the $p(\zeta|s, a, s')$ which is similar to $p(\eta|a, x, s)$ in our case. However, for us, we use the sample η and generate a, x from it. Hence, we do not need to learn the model $p(\eta|a, x, s)$. Instead, we can just record the value of η when we are collecting the training samples.

The last part of the algorithm is to make the gradient estimation not depend on the samples collected by $\pi(a, x|s)$, as the policy is constantly changing. We use the replay buffer from DDPG to solve this issue and turn our algorithm into an off-policy algorithm to improve sample efficiency.

Experiments

We conducted our experiments using the Platform domain from (Masson, Ranchod, and Konidaris 2016) (See Figure 2). In this domain, we control the agents (cyan block) to jump across several platforms while avoiding enemies (red blocks) and falling off the platforms. This domain has three discrete actions to choose from: run, jump, and leap. A jump moves the agent over its enemies, while a leap propels the agent to the next platform. Each of the actions take one parameter which determines the speed along the x-axis. More details of the domain can be found in the original paper.

We implemented the Parameterization Action DDPG[†] (PADDPG) algorithm from (Hausknecht and Stone 2015) as our comparison baseline which is considered as the state of the art. Specifically, we implemented the PADDPG algorithm following the settings and parameters from the original paper except for the size of the hidden layers. In the original paper, PADDPG used a huge network with four hidden layers with size $\{1024, 512, 256, 128\}$, which is rare in DRL community for tasks with continuous signals. We followed the DDPG paper (Lillicrap et al. 2015), which used two hidden layers with sizes $\{400, 300\}$ for the neural networks. We also implemented their invert-gradient trick, as they claimed that this was the only way to make the learning work in a bounded parameter space.

For our PATRPO agent, we adopted the setting from (Duan et al. 2016), where we had three hidden layers with sizes $\{200, 100, 50\}$ for the policies. We used ReLU for activation, and Softmax and Tanh for the output layers of the discrete action policy and continuous parameter policy respectively. For our PASVG(0) agent, we also used neural networks with two hidden layers of sizes $\{400, 300\}$ and ReLU for activation. For the output layer, we used Gumbel Softmax for the discrete action policy and Tanh for the continuous parameter policy.

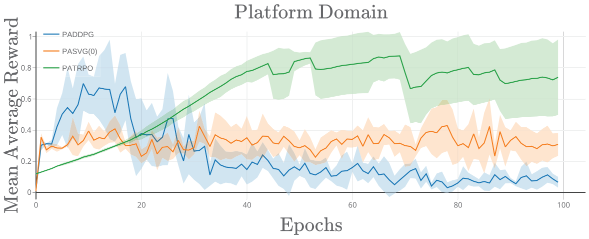


Figure 3: Comparison on Platform domain of three learners. The x-axis shows the training epochs. The y-axis shows the average reward. Solid lines are average value over five random seeds. Shaded regions are standard deviation.

We trained all the agents using 100 epochs with 10000 samples per epoch. For the online method, we had a replay buffer with size 10^7 and we did not start the training until we had 10^4 samples in the replay buffer, which is a standard setting in DRL experiments. We used 0.005 as the step size for PATRPO agent and 10^{-3} and 10^{-5} as the learning rate for the value function and policies respectively for the PASVG(0) agents. We fixed the temperature to 1.0 for the Gumbel-Softmax layer and kept it for the entire training process.

The experiment results are shown in Figure 3. The plot of PADDPG is very interesting: we found that it can learn to successfully finish the game at an early stage of learning, but would quickly unlearn that policy and converge to something else.

We then noted that, although we are using Tanh to bound the output of our parameter policy, which corresponded to the squash-gradient method in (Hausknecht and Stone 2015), we managed to make it work for our methods, which suggests that there are more training options than the invert-gradient method suggested by (Hausknecht and Stone 2015). Our PATRPO method achieved the best performance among all three learners, and unlike PADDPG learner, it maintained its performance after obtaining its best learned policy. The PASVG(0) learner converged to a local optimum with average reward of around 0.4. By examining the game, we found that this corresponded to avoiding the first enemy but failing to land on the second platform. One of the possible reasons was that the learner was conducting joint-learning, which is very similar to cooperative multiagent learning, and thus may converge to a local optima.

We further investigated this joint-learning issue by trying two different step sizes for the PATRPO agent. Figure 4 shows the result of using larger step size parameters. As we can see, both of the agents can achieve a good performance in relatively short period of time with much lower variance. But once they learn the optimal policy, their performance starts to drop and the variance becomes much larger. However, PATRPO still manages to maintain a decent performance which is far better than the PADDPG algorithm. This shows that a more stable method is desirable for learning in

[†] This is the DDPG algorithm for parametereized action spaces, not to be confused with the DDPG algorithm from (Lillicrap et al. 2015) for continuous control.

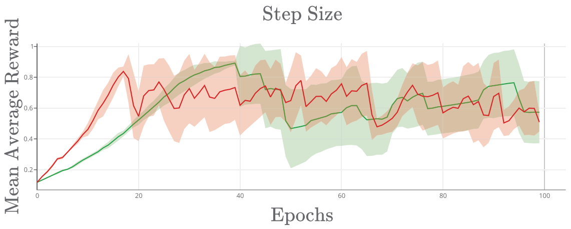


Figure 4: Different step size parameters for PATRPO agents. $\delta = 0.05$ in red and $\delta = 0.01$ in green.

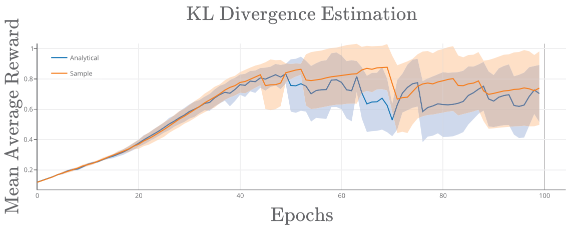


Figure 5: Comparison on the Platform domain for different KL-Divergence estimation methods.

the parameterized action space.

Last, we conducted an experiment using different techniques for estimating the KL divergence in PATRPO. The experiment as illustrated in Figure 5 showed that none of them makes much of a difference in this small domain.

We also tested our algorithm in the HFO domain introduced by (Hausknecht and Stone 2015). In this domain (Figure 6), we controlled an agent to score a goal. For the simplicity, we did not have a goalie. We had three actions in this domain, **dash** with parameters *power* and *direction*, **turn** with parameter *direction* and **kick** with parameters *power* and *direction*. Thus, different actions required different numbers of parameters. For our agents, if we outputed more parameters than we actually needed, we just took the first part of the output and ignored the remainder. This domain had a 59-dimensional state space, which was much larger compared to the platform domain, and thus we trained our agents using larger neural networks and with more samples. Due to time constraints, we only trained our PATRPO agent and PADDPG in this domain for 100 epochs with 50000 steps per epoch. We used three hidden layers with size $\{400, 300, 200\}$ for both PATRPO and PADDPG agents.

Figures 7 shows the result on this domain. As we can see, again, the PATRPO agent achieved stable performance in this domain while PADDPG demonstrated a large variance in its performance. We also note that the performance of the PADDPG algorithm is far worse than in the original paper. One of the possible reasons for this is due to the difference in the neural network sizes. But since our PATRPO agent can achieve stable learning in this domain with a much smaller neural network, this suggests that a large neural network is not necessary in this domain.

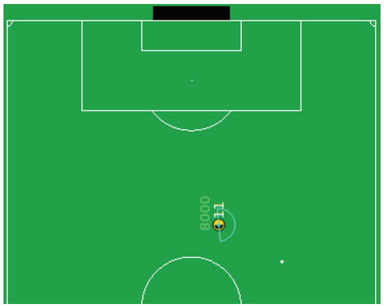


Figure 6: An example of Half Field Offense Domain, with no goalie.

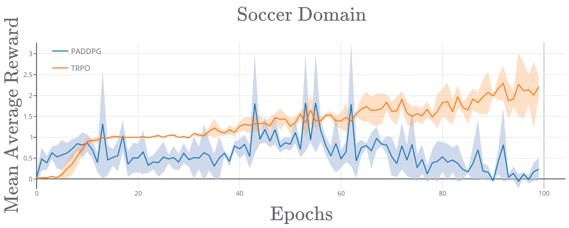


Figure 7: Comparison on Soccer domain for PATRPO and PADDPG agents on three different random seeds.

Conclusion and Future Work

We presented two algorithms for learning effective control in parameterized action space. We demonstrated that our method can learn better policy in these setting compared to PADDPG method. However, we found that learning could be unstable due to the joint-learning between the discrete action policy and parameter policy. An interesting future direction would be to find more stable methods for this domain. We would like to study these methods in the context of more complex domains (in soccer for example) particularly involving more agents.

Acknowledgments

The research in this paper was conducted with the support of research infrastructure developed under NSF grant 1727303.

References

Baram, N.; Anschel, O.; Caspi, I.; and Mannor, S. 2017. End-to-end differentiable adversarial imitation learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, 390–399.

Baxter, J., and Bartlett, P. L. 2001. Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research* 319–350.

Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; and Abbeel, P. 2016. Benchmarking deep reinforcement learning for continuous control. In *Proceedings of The 33rd International Conference on Machine Learning*, 1329–1338.

Hausknecht, M., and Stone, P. 2015. Deep reinforcement learning in parameterized action space. *arXiv preprint arXiv:1511.04143*.

Heess, N.; Wayne, G.; Silver, D.; Lillicrap, T.; Erez, T.; and Tassa, Y. 2015. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems*, 2944–2952.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Jie, T., and Abbeel, P. 2010. On a connection between importance sampling and the likelihood ratio policy gradient. In *Advances in Neural Information Processing Systems*, 1000–1008.

Kakade, S. M. 2002. A natural policy gradient. In *Advances in neural information processing systems*, 1531–1538.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Maddison, C. J.; Mnih, A.; and Teh, Y. W. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Maddison, C. J.; Tarlow, D.; and Minka, T. 2014. A* sampling. In *Advances in Neural Information Processing Systems 27*, 3086–3094.

Martens, J. 2010. Deep learning via hessian-free optimization. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 735–742.

Masson, W.; Ranchod, P.; and Konidaris, G. 2016. Reinforcement learning with parameterized actions. In *AAAI*, 1934–1940.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Mordatch, I., and Abbeel, P. 2017. Emergence of grounded compositional language in multi-agent populations. *CoRR* abs/1703.04908.

Pearlmutter, B. A. 1994. Fast exact multiplication by the hessian. *Neural computation* 6(1):147–160.

Rachelson, E.; Fabiani, P.; and Garcia, F. 2009. Timdppoly: An improved method for solving time-dependent mdps. In *Tools with Artificial Intelligence, 2009. ICTAI'09. 21st International Conference on*, 796–799. IEEE.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 1278–1286.

Schulman, J.; Levine, S.; Abbeel, P.; Jordan, M.; and Moritz, P. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, 1889–1897.

Sharma, S.; Lakshminarayanan, A. S.; and Ravindran, B. 2017. Learning to repeat: Fine grained action repetition for deep reinforcement learning. *arXiv preprint arXiv:1702.06054*.

Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; and Riedmiller, M. 2014. Deterministic policy gradient algorithms. In *ICML*.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489.

Sohn, K.; Lee, H.; and Yan, X. 2015. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, 3483–3491.

Stone, P.; Kuhlmann, G.; Taylor, M. E.; and Liu, Y. 2006. Keepaway soccer: From machine learning testbed to benchmark. In Noda, I.; Jacoff, A.; Bredenfeld, A.; and Takahashi, Y., eds., *RoboCup-2005: Robot Soccer World Cup IX*, volume 4020. Berlin: Springer Verlag. 93–105.

Sutton, R. S.; McAllester, D. A.; Singh, S. P.; and Mansour, Y. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, 1057–1063.

Vinyals, O.; Ewalds, T.; Bartunov, S.; Georgiev, P.; Vezhnevets, A. S.; Yeo, M.; Makhzani, A.; Küttler, H.; Agapiou, J.; Schrittwieser, J.; et al. 2017. Starcraft ii: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*.

Walker, J.; Doersch, C.; Gupta, A.; and Hebert, M. 2016. An uncertain future: Forecasting from static images using variational autoencoders. In *European Conference on Computer Vision*, 835–851. Springer.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Zamani, Z.; Sanner, S.; Fang, C.; et al. 2012. Symbolic dynamic programming for continuous state and action mdps. In *AAAI*.

The Design of the User Experience for Artificial Intelligence

Usability Issues and Guidance for Flexible Execution of Procedural Work

Dorrit Billman,[1] Debra Schreckenghost[2]

[1]San Jose State University @ NASA Ames Research Center, [2] TRACLabs, 16969 N. Texas Ave, Suite 300, Webster, TX 77598
dorrit.billman@nasa.gov, schreck@traclabs.com

Abstract

We believe user experience in complex work domains is shaped by the effectiveness of technology in jointly accomplishing work goals. Function allocation between humans and smart technology is an important part of effectiveness, in turn, an important contributor to user experience. We study operation of equipment for the International Space Station, using procedure automation with flexible function allocation. We discuss automation goals, their impact on suitable function allocation, and the role of flexible function allocation. We offer examples and propose guidelines.

Bios & Research Background

Dorrit Billman is a cognitive scientist with background in learning and in human factors of tools for complex cognitive work. She has researched tools for ecological modeling, collaborative intelligence, and mission planning. Debra Schreckenghost is a Senior Scientist at TRACLabs. She has conducted research in the areas of adjustable autonomy, human interaction with automation, and real-time performance assessment of robots and automation.

Our research investigates design of procedures, procedure automation software, and particularly, how users interact through software to operate complex equipment. From observing users working with a variety of function allocation policies, we have identified a number of issues around teaming with smart technology.

Source Domain

We investigate user-automation interaction in accomplishing technical work, specifically, operation of habitat systems for the International Space Station. Currently, work is carried out manually by astronauts following written procedures, with support from Mission Control. Future operations will rely more on automation, and reduce dependence on Mission Control. Our studies investigated users carrying out (simulated) operations using the PRIDE procedure automation software (Billman, Schreckenghost, & Billinghurst 2015). Technically, the PRIDE software is a knowledge-based system that uses a hierarchical task language to automate system monitoring and control (Schreckenghost, et al 2008). Behaviorally, it captures the knowledge developed at the organizational and individual level for operating equipment safely and effectively, and allows both automatic and teamed execution of procedures.

In work domains, particularly safety-critical, technical domains, the key elements of user experience are effectiveness and efficiency in accomplishing work goals. Contributions of the technology to these goals are more typically described as usefulness and usability. However, the centrality of these aspects to “user experience” cannot be over-emphasized; people’s experience is deeply shaped by their ability to work effectively.

We believe that ability to team with and use “smart technology,” to accomplish goals is important in many domains. We expect there will be important cross-domain commonalities, particularly in the design issues and trade-spaces, while appropriate solutions will vary to meet domain needs. We discuss the issues and design trade-spaces concerning how work should be coordinated and distributed across human and artificially intelligent entities. We invite discussion of domain similarities and differences.

Work Distribution in Procedural Domains

Procedural work domains are those where work is meaningfully organized around individuated, discrete actions, and correct ordering of the actions is important and somewhat standard. Typically, the importance of sequencing comes from physical or process constraints, but it could also be driven by benefits from establishing a standard order. Procedural work contrasts with continuous control, which dynamically adjusts control parameters to maintain a control policy. It may also contrast with domains where work is composed of discrete actions, but sequence is less important or less standard; creative work might provide examples. We do not imply that all work in “procedural domains” involves procedures, but that procedures can centrally contribute to accomplishing work goals. We focus on mixed initiative procedure execution, involving one person who is working concurrently with automation on one or more procedures.

Supporting procedural work involves several design considerations. The foundational design guidance is derived from an analysis of the work domain, including characteristics of the work goals, users, technology, and work context. Without a sound understanding of the work needs, procedure automation software is unlikely to be helpful. A second source of design guidance and constraint comes from the design of procedures appropriate to the domain. The high-level goal of a procedure is typically to effect productive change, or to mitigate the effects of anomalies. Additional goals may include safety and risk management, efficiency, consistency, and appropriateness for execution by available human operators with available technology. Where automation is used, it is a valuable strategy to use procedures to organize how the automation contributes. A third source of design guidance comes from the capabilities of the entities available to execute these procedures and the reasons for using automation to execute the work. In procedural domains automation is often introduced to execute component actions originally executed only by humans; in such cases, automation is introduced into an existing work process. Alternatively, automation may be used for new tasks, tasks that people cannot do themselves, or cannot do safely. If both the human and the automation are capable of carrying out certain units of work, whether a whole procedure or a subset of actions within it, then there are choices in how work should be distributed among humans and automation and a function allocation policy is needed.

Design of a function allocation policy should be informed by the overall purpose of automating work. Design of function allocation needs to specify the units of work that can be allocated (e.g. an action, all actions of a type, a whole procedure, etc.). In addition, allocation may concern who *does* the unit of work (e.g. manually plugging a device in vs. automatically switching on current), or may concern who *selects* an action for execution (e.g. a user selects a command to automatically change a component). If there are function allocation choices, there may also be flexibility in which allocation is chosen. This may serve a domain need for flexibility in the work more generally.

Goals Served by Function Allocation

Function allocation, or distribution of work across team members, should be guided by the objectives important in the domain. Function allocation can serve multiple, sometimes conflicting objectives, beyond the overall goal of the specific procedure. Some goals favor maximal automation, such as minimizing human slips in initiation and execution of actions or protecting humans from hazards. Some goals favor maximal allocation to humans, as when conditions for actions are hard to specify at time of design or when human capabilities are hard or costly to automate. Further, some goals favor mixed allocation, such as requiring users to issue critical commands while automatically verifying that commands have the intended effect, keeping humans in the loop to maintain situation awareness, or gaining assurance from cross-checking between human and automation assessment.

Flexible function allocation enables the user to select which of these allocation policies best matches current operational goals. Adjusting function allocation may help i) maximize usable human time, ii) aid human monitoring, learning, skill maintenance, engagement, or understanding of system operation, and iii) balance workload and safety considerations for changing conditions, resources, or goals.

Frequently, multiple goals, possibly in competition, are in play. In our domain, a key goal is effective use of human time. Astronauts have high workload and limited time, so freeing time is an important goal for automation use.

We are particularly interested in *flexible function allocation*. Effective operations can require flexibility in what is done and who does it. For example, in current practice changes to procedures may be required when equipment ages or is modified. Flexible allocation would allow a shift from automatic to manual execution for this adaptation. In addition, certainty that a procedure is appropriate and will have the intended effect may drop if a new component is installed or systems are configured in a nonstandard way. In such cases it may be particularly valuable to have a person closely monitoring and manually initiating procedure actions. Conversely, as equipment, context, and procedure become well-established, automatic execution may be most effective. The state of the human operator will also influence the relative benefit of automated vs manual execution.

Allocation Goal: Freeing Human Time

In our studies we were interested in how design of function allocation affects availability of useful human time. While improved quality of work is a reason for human or human-automation teaming, greater efficiency is also a motivation. With the types of procedures executed in our domain, much of the execution time is a function of the equipment operated, not the operator actions. When the person does not need to be actively involved, this can free up operator time for manual work in parallel activity.

Team-work costs vs. task-work benefits. As with human collaboration (McGrath, 1984), time on teamwork, such as coordination, reduces time available for task-work. In general, the team-work added should be less than the task-work reduced. In our domain, team-work added time at handoffs and automation failure. There may be other costs, such as awareness or workload. For example, users sometimes deviated from the allocation plan when it stipulated frequent handoffs. When a small block of automated actions occurred between actions to be done manually, users sometimes did these to-be-automated actions manually, apparently to avoid the team-work cost of task switching.

Guidance: Substantial costs in human time occur at task handoffs between manual and automated execution. Primary benefits to freeing human time come from contiguous stretches of automatable actions. Proportion or number of automated actions may be less consequential than their distribution. There may be a tradeoff between maximizing the amount of automated work and minimizing handoffs.

One strategy is to allocate commands to a person and verification of command effects to automation. This can increase team-work costs due to frequent handoffs between humans and automation. If frequent handoffs are required, the handoff cost should be minimized in other ways, e.g., design the automation to automatically resume when manual actions are complete.

Handoff frequency. Minimizing the handoff frequency and maximizing the time span of automated actions between handoffs is often beneficial. The longer the user has between manual actions, the larger the block of time available for other work. Reducing handoffs reduces time spend on team-work. For example, when users encounter short stretches of automatable actions between manual ones, they often did the automatable actions manually to eliminate the time-costs of the skipped handoffs.

Guidance: When condition dependencies permit, the actions in procedures intended for human-automation execution should be grouped and ordered into Manual Only action sets and Automatable action sets. This can reduce the frequency of required human-automation handoffs. Note that this order principle may compete with others, such as grouping actions together that accomplish the same goal.

Interruption. Interruptions disrupt human work at handoffs, for user approval, procedure completions, or anomalies. Interruption from team-work should minimize impact on task-work. This may be mitigated by designs conveying, information about importance and urgency. In PRIDE, the automation pauses when a manual-only action is reached, for user approval, an automated action fails, or the procedure completes. While the user should be made aware of all these pauses, there usually is more urgency in responding to a manual action requesting approval, or for the failure of an automated action.

Guidance. The automation interface should draw the operator's attention when it stops doing task-work and indicate why it stopped. The saliency of these notices should be designed to balance the urgency and importance of information with the potential for unnecessary distraction.

Situation awareness with automation. The goal of reducing human time interacting with automation performing procedural work must be balanced against the need for the human to maintain awareness of the domain systems and automation behavior. Procedures that are performed with less human involvement can require providing information about both the system states and the task completion to orient the user when intervention is required.

The use of procedures as the basis of automation is intended to make the automated actions more transparent to the user. We found it useful to provide a user interface that annotated the procedure text with information about what actions had been taken, what states had been changed, and what action the automation is currently performing.

Guidance. Situation awareness and problem solving with procedures may be improved by including in the procedure information about what system states or environmental conditions the procedure actions are intended to change and what system states or environmental pre-conditions are assumed to be true before actions are executed.

An automation system also might provide information on activity in multiple tasks. This could inform awareness of probable workload and could guide changes in order or timing of requests for manual actions. Prioritizing actions across tasks is research issue in aviation autoflight systems.

Allocation Goal: Flexibility

If procedure designers could perfectly predict the conditions and goals in force when a procedure would be executed, it would be possible in principle to specify an optimal allocation policy. However, in any complex domain complete forethought is unlikely. Flexible allocation is more valuable when the domain is more complex, less understood, has lower degrees of predictability or when normal operations include high variability in system and user behavior. These traits characterize technical operations in dynamic environments. Even anticipated variation can have a high planning cost. Flexible allocation is less valuable in highly predictable domains, where standardization is possible and useful, e.g., in aiding handoffs, or collecting data about system performance during operations.

Scope and nature of flexible allocation. Function allocation may be done by the designer when a procedure is written; by the operator in advance of procedure execution (allocation replanning); or by the operator during execution (reactive allocation). Flexible allocation is only possible for actions that can be done manually or automatically. If these are few, flexible allocation may not be worthwhile. To illustrate benefit, some tasks can require reallocation when components used in the task have been replaced or repaired, and the user should execute relevant actions manually to ensure the new component responds normally.

Guidance. Prior to design, benefits of flexible allocation, should be assessed relative to automation goals. Flexible allocation adds complexity for the operator, as well as development costs, so there should be commensurate taskwork benefit. Avoid adding complexity without benefit.

Methods for changing function allocation. Where flexibility is beneficial, the operator should have methods for planned and reactive allocation. Planned allocation might be used to standardize routine allocations, e.g., automating all verify instructions. Reactive allocation can be used to 'short cut' a local inefficiency, e.g., doing a few actions manually to avoid the overhead of shifting to automatic. Methods for reactive and planned reallocation will differ e.g., for reactive reallocation, we provided a simple way to stop automation when the currently executing action completes. Multiple methods were provided for planned reallocation, including reassignment of actions throughout a procedure.

Users may need to coordinate interactions among procedures, particularly if procedures can be run in parallel.

Guidance. If allocation during execution is valuable, methods should inform the operator which actions and procedures can be re-allocated. The team-work costs of reallocation should be less than the task-work benefit. Determining the goodness of flexible function allocation requires understanding the intended benefit and assessing whether this benefit was found. If freeing human time is the goal, interaction methods should reduce time needed from humans. Other goals, such as human safety, may require other methods, such as cross-checking.

Transparency of function allocation. The types of work units that can be flexibly allocated should be clear. In our case, a procedure as a whole, a functional set of actions (a step), and a single procedure action could each be flexibly assigned. Any unit could be executed manually, but only some automatically. Thus, if the user selected automatic execution for a whole procedure, automation stopped when a manual-only action was encountered.

One procedure interface we evaluated required the user to infer which actions could be automated (by recognizing that automatable lines required a telemetry read-out or command button). A number of users had difficulty with this. After a redesign, each procedure action was explicitly marked as manual only (M) or automatable (A). Using this new interface, very few users had difficulty in knowing which actions would execute automatically, which manually, and where automation would pause. As a second example, after a manual action the automation might resume on its own, or require an explicit user action to resume.

Guidance. The procedure automation interface for executing flexible allocation plans should show users which actions are currently marked for automated versus manual execution and what allocation choices are available. The handoffs during execution should be clear, showing when the automation will act, and when user action is expected.

Conclusion

In technical work domains, we suggest that a core driver of positive user experience when using automation is effectiveness in accomplishing the joint work. We discuss the issues that emerge in designing 'team-work' as well as 'task-work' for mixed initiative operation of equipment. A key element is function allocation. We suggest the following design guidance will be helpful for managing the distribution of work among humans and smart technical systems, in multiple work domains.

- Goals of Automation. The goals of automating should guide design of the interaction to control work allocation. Interaction needed for flexible function allocation has costs in development and use, which should be weighed against intended benefits of flexibility.
- Automation Transparency & Directability. Automation technology should transparently show how work is and can be distributed. Such technology should provide directability of how work is distributed, thus providing flexible allocation.
- Multiple Contexts. Automation is more adaptable if work-allocation can be assigned in various contexts, such as automation technology design, procedure writing and work design, operations planning, and operations execution including parallel execution of tasks.
- Alignment with Structure of Human Work. To improve flexibility, the representation of work used in function allocation should be aligned with how people think of and do the work. Allowing work to be allocated at multiple levels of abstraction and granularity increases adaptability of the automation; work may be distributed at the level of an individual action, a functional group of actions, a simple procedure, or a complex procedure with nested sub-procedures.

Acknowledgements

Research was supported by the National Space Biomedical Research Institute (NSBRI) through NASA NCC 9-58.

References

Billman, D.; Schreckenghost, D.; and Miri. P. 2014. Assessment of Alternative Interfaces for Manual Commanding of Spacecraft Systems: Compatibility with Flexible Allocation Policies," *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 58, pp. 365–369.

McGrath, J.E. 1984. A Conceptual Framework for the Study of Groups, in *Groups: Interaction and Performance*, Englewood Cliffs, NJ: Prentice-Hall.

Schreckenghost, D.; Bonasso, R.; Kortenkamp, D.; Bell, S.; Milam,T.; Thronesbery, C. 2008. Adjustable Autonomy with NASA Procedures. *9th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Pasadena, CA.

Building Bridges: A Case Study in Structuring Human-ML Training Interactions via UX

Johanne Christensen, Benjamin Watson
Department of Computer Science
North Carolina State University
{jtchrist,bwatson}@ncsu.edu

AJ Rindos, Stacy Joines
IBM
{rindos,joines}@us.ibm.com

Abstract

With the increasing ubiquity of artificial intelligence and machine learning applications, systems are emerging that require non-ML experts to interact with machine learning at the training step, not just the final system. These users may not have the skills, time, or inclination to familiarize themselves with the way machine learning works, so training systems must be developed that can communicate the necessary information and facilitate effortless collaboration with the user. We consider how to utilize techniques from qualitative coding, a human-centered approach for manual classification, and build better user experience for ML training.

Introduction

Technology has always been designed to assist humans in completing their tasks. As it advanced, particularly in computing, the types of tasks that technology could address shifted from simple (performing calculations or formatting text) to more complex (language understanding and image recognition). While complex AI applications are on the rise, their deployment still requires significant human effort, often by domain experts who have little understanding of how AI works. Thus when training a machine learning (ML) model, domain experts must work with ML experts, making ML solutions for many tasks too costly or simply infeasible. If we are to realize the full potential of machine learning (and, by extension, AI), we must begin building systems that can be deployed or at least improved by end-users.

User experience (UX) seeks to create products that are not only useful and usable, but also motivating and pleasurable. Toward this end, UX practitioners employ a human-centered practice that we believe is essential for building AI or ML solutions useful by domain experts without ML familiarity. Such solutions must communicate effectively with users about their domain, without bogging them down in ML detail.

Classifier Training for non-ML Experts

Even when ML expertise is unnecessary or can be minimized, the requirement of extensive domain expertise can be a significant barrier to adoption. In specialized domains such as law or finance, expert time is expensive. Methodologies that reduce the required number of labeled examples may not be enough: very large datasets can still make the labeling task impractical.

Nevertheless, complete automation of the training process is not only infeasible, but also inadvisable. Human oversight creates trust in the model once it is deployed.

Qualitative Coding

Qualitative coding (QC) (Saldaña 2015) is a manual classification method commonly employed in the humanities and behavioral sciences to extract meaning from data such as text, imagery, and video. The subset of QC called grounded theory particularly has much in common with traditional machine learning (Muller et al. 2016), especially in the way that theories (or models) are built from the data. With this in mind, we posit that QC can form an interface for ML training, facilitating interaction and structuring dialogue around the data for analysis. QC may be ideal for building an ML platform that domain experts can comfortably use, without significant ML expertise.

(Shneiderman 1982) argued that direct manipulation in a desktop interface provides a better experience for non-expert computer users. Similarly, we believe that QC supports direct manipulation of data, creating an intuitive yet still efficient experience. QC practitioners commonly use note cards or post-its to represent data, physically grouping post-its to cluster data points. Attaching meaning to movement in the ML training interface should increase communication bandwidth, allowing domain experts to communicate about their data not only linguistically, but also behaviorally.

Insights from the Field

We have recently conducted interviews with developers building custom interfaces for clients to train classification models on unstructured text documents. In these systems, out of the box, the ML software is able to classify documents with limited accuracy. For improved accuracy, domain experts must train the system further by labeling data.

Our interviews have surfaced several pain points during ML training. Interaction quickly becomes repetitious, with experts providing feedback about the classification of several data items, iteration after iteration. Developers observed that users find labeling documents burdensome.

One method of mitigating this tedium is to simplify the feedback task: instead of asking how the document should be classified, users might indicate only whether the current label is correct. Yet this requires a system that can:

- identify documents that likely should have the current label
- decide which of those documents need manual labeling, ie. which examples, once manually labeled, will most improve the model. Indeed, such a capability might also be used to reduce the number of feedback task iterations domain experts must perform.

When relevance is more nuanced, answering a yes/no question about correctness might be too simple to produce a good model. Ranking of relevance might be a good compromise, but as feedback complexity grows, so does subjectivity between coders. How should a model account for the possibility that two domain experts may have differing ideas on relevance? In QC methodology, memos allow coders to compare notes on why they picked certain codes, and statistical intercoder agreement scores measure overall cross-coder consistency.

Another way of reducing the tedium of feedback is to vary the task, and spread it across multiple expressive modalities (e.g. language, behavior, vision and sound). We believe that the movement and highly visual nature of QC coding and its data displays will be quite helpful in this regard.

Nearly all developers mentioned the difficulty of communicating the confidence the ML model has in its classifications to non-ML experts. Domain experts should prioritize feedback about high confidence classifications that are wrong, and about lower confidence classifications in general. Domain experts often misinterpreted confidence scores, believing that they were being provided with classifications already known to be incorrect. This often led them to provide inaccurate feedback.

The obvious solution is to train users about confidence scores. However, a domain expert who specializes in a field that does not regularly utilize probabilities neither needs nor wants to learn what confidence scores mean. Instead, as one developer observed, rather than asking domain experts to focus on certain documents based on classification confidence, we might highlight those documents in display (and perhaps filter out documents not needing attention, again reducing feedback task iterations).

How can QC structure interaction with ML?

Our goal is to increase the utility and accessibility of ML algorithms by making interaction with them understandable, efficient and engaging enough to allow domain experts to train them, and to explain their results to their peers. To achieve this, we will hide algorithmic detail with QC-based interaction focused around data, and with ML classifiers treated as collaborative coding partners.

Below, we sketch the specific challenges of human-ML interaction and explore how QC addresses (Nielsen and Molich 1990)'s usability heuristics.

- *Recall*. Manually creating an ML training set is difficult, but evaluating much larger ML algorithm results is daunting, requiring users to recall and navigate connections between dozens of labels and thousands of examples (or more). QC-based codebooks (label indexes), data displays (using note cards and post-its), memos and histories help domain experts remember and navigate through such large collections of information.
- *Error correction*. Errors during model building can come from either the human or the ML. QC relies on iterative reflection to correct errors. Data displays provide the context in which errors can be identified, while using displays to communicate how well training examples cover the data, how well training labels (or classes) fit data features, and examples that significantly influenced the classifier provide multiple opportunities for finding and resolving errors.
- *Iteration*. ML training requires extensive iteration, and evaluating the results of each iteration is difficult, particularly for domain experts. QC supports manual iteration with improved measures of coding accuracy, and a focus on key data examples. Labeling in iterations also breaks the task into more manageable pieces, mitigating fatigue, attention drift and stress for the user.
- *Collaboration*. For reasons of efficiency, reliability and trust, classification in many applied settings is intensely collaborative. Collaboration in QC is inbuilt, supporting the dialog of live partners with displays and intercoder agreement measures. The ML is considered an additional partner, so providing human-ML interaction on par with human-human interaction is essential.
- *Efficiency*. ML often assumes that users already know how to label the training set, how to label it efficiently, and that they will label examples one data dimension at a time. QC includes grounded coding, with codes emerging as researchers encounter the data; and simultaneous coding, with researchers attaching multiple codes to each data item. These techniques also support users learning the task as they perform it, as it is flexible enough that novices to the training task itself can effectively interact with the system from the start. Even during practice, users can contribute to training, even if some of their input needs to be changed later.
- *Interaction*. Many domain experts structure data by pushing paper representations of their data into piles and many QC researchers still prefer this manual coding experience to the digital one offered by qualitative coding software. Current ML systems cannot support such a natural interaction. We envision tabletop and wall displays that reproduce this intuitive experience to allow domain experts to create training sets and evaluate classification results.

Structuring a Dialogue Beyond Words

In a human-machine partnership, what should the interaction look like? QC methodologies are often described as structuring a dialogue around and about the data. Extending that metaphor into a design consideration provides a strong foundation for building ML software that helps non-expert users work in partnership with the machine to complete their desired task.

Although it is common to consider a dialogue as an exchange of words, communication around a subject need not be linear, synchronous, or restricted to language. For example, using visual representations to summarize data or non-verbal behaviors to convey ancillary information are alternative techniques that can carry a dialogue without human (or computer) language.

With this in mind, we consider what a dialogue about training a classifier might look like. Crucially, it is vital that the system is structured so that the machine communicates at a human level, accommodating users at all levels of technical expertise. In this setup, the burden of driving the conversation and managing the task remains on the system, but the user remains the ultimate authority, with the final say should there be a dispute. Table 1 shows a possible task breakdown between the human and machine partners.

We imagine an interface where the system continually communicates the state of the model in training. This could be via visualizations of the progress of the model as it approaches a trained state. In addition to providing information to the user, a complete solution for communication requires a system that can properly decode the user's state to fully understand the user's actions. For example, if the user hesitates when labeling a data point, the system might learn that this behavior means the label should be applied with a lower confidence score.

Human Partner	ML Partner
labels docs manages the knowledge corrects errors	gives feedback on model manages the data finds possible errors manages task breakdown learns from user's actions

Table 1: Task breakdown between Human and System

Consider an ML classifier that groups data items into two categories: relevant and irrelevant. In a tabletop display that reproduces QC interaction, a domain expert is training the ML. A visualization shows that the ML model has not yet classified a third of the data, and that another 5% of the data is poorly fitted. Directly in front of the expert, dozens of data items are represented by cards. Two piles of cards are labeled "relevant" and "irrelevant," another "unlabeled," and a fourth "revisit."

The expert drags one of the unlabeled cards to the center, where it expands, showing additional detail. She considers for a moment, then swipes the card rapidly toward the "relevant" pile. The card spins and curves on its way to the pile where it settles in with an audible plop, and the visualization updates as a result, with only a quarter of the data still unclassified. The expert then brings another unlabeled card to the center. This data item is more difficult, and she consults her own and others' memos in the QC codebook before swiping the card to the "irrelevant" pile. The classifier notes the expert's hesitation and marks the labeling of this data item as "uncertain."

Bridging the Disconnect between Humans and Machines

Understanding how humans communicate with machines is key to building effective interactive systems (Suchman 1987), like those being developed for ML classifier training. UX's human-centered approach is particularly valuable in this regard. For example, the human tendency to anthropomorphism can be leveraged to enrich human-machine interactions (Levillain and Zibetti 2017)(van Allen 2017). By treating human-machine interaction as a form of human-to-human communication, we might improve interaction, particularly for non-technical users. Today's technology may finally be enabling systems that realize this vision and the vision of affective computing (Picard and Picard 1997): perceiving, understanding and expressing – communicating – with users not just by language and example, but by behavior and emotion. This sort of interaction is rapidly becoming necessary as human machine dialog enters all phases of our daily lives and indeed our lifetimes.

Acknowledgements

This work was partially funded by an IBM Faculty Award.

References

Levillain, F., and Zibetti, E. 2017. Behavioral objects: the rise of the evocative machines. *Journal of Human-Robot Interaction* 6(1):4–24.

Muller, M.; Guha, S.; Baumer, E. P.; Mimno, D.; and Shami, N. S. 2016. Machine learning and grounded theory method: Convergence, divergence, and combination. In *Proceedings of the 19th International Conference on Supporting Group Work*, GROUP '16, 3–8. New York, NY, USA: ACM.

Nielsen, J., and Molich, R. 1990. Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 249–256. ACM.

Picard, R. W., and Picard, R. 1997. *Affective computing*, volume 252. MIT press Cambridge.

Saldaña, J. 2015. *The coding manual for qualitative researchers*. Sage.

Shneiderman, B. 1982. The future of interactive systems and the emergence of direct manipulation. *Behaviour & Information Technology* 1(3):237–256.

Suchman, L. A. 1987. *Plans and situated actions: The problem of human-machine communication*. Cambridge university press.

van Allen, P. 2017. Reimagining the goals and methods of ux for ml/ai. *AAAI Spring Symposium Series*.

User Interfaces and Scheduling and Planning: Workshop Summary and Proposed Challenges

Richard G. Freedman
University of Massachusetts Amherst
freedman@cs.umass.edu

Tathagata Chakraborti
Arizona State University
tchakra2@asu.edu

Kartik Talamadupula
IBM Research
krtalamad@us.ibm.com

Daniele Magazzeni
King's College London
daniele.magazzeni@kcl.ac.uk

Jeremy D. Frank
NASA Ames Research Center
jeremy.d.frank@nasa.gov

Abstract

The User Interfaces and Scheduling and Planning (UISP) Workshop had its inaugural meeting at the 2017 International Conference on Automated Scheduling and Planning (ICAPS). The UISP community focuses on bridging the gap between automated planning and scheduling technologies and user interface (UI) technologies. Planning and scheduling systems need UIs, and UIs can be designed and built using planning and scheduling systems. The workshop participants included representatives from government organizations, industry, and academia with various insights and novel challenges. We summarize the discussions from the workshop as well as outline challenges related to this area of research, introducing the now formally established field to the broader user experience and artificial intelligence communities.

1 Introduction

One of the earliest areas of research within artificial intelligence (AI), planning and scheduling (PS) studies the selection of sequences of actions to accomplish tasks. This field broadly encompasses studying the representation of knowledge and information, such as representing goals, tasks and constraints, and developing problem solvers using search methods and heuristics. Automated planning and scheduling technologies have been used in applications ranging from supply chain management to robotics to space mission planning. Many of these technologies were designed by members of the International Conference on Automated Planning and Scheduling (ICAPS) community.

Although many useful techniques and formulations for problem definitions and solutions have been devised by the PS community, the algorithms and methods are often not very friendly to users outside the research community. With some exceptions, capturing knowledge and plan display is done via text files, without any guide or visual aids. This approach is suitable for researchers, but not users of planning and scheduling systems, which is a barrier to wider adoption of innovations in the field. In particular, the utility of PS technology for those outside the community is often constrained by the user interface (UI) design. Members of the ICAPS community as a whole have noted that application developers are overlooking automated PS technologies in domains where it should be used, and the lack of good UIs may be one reason for this.

Recent advances in interfacing modalities such as natural language processing (Munteanu et al. 2017) and augmented reality (Chi, Kang, and Wang 2013) call for an investigation of novel ways to facilitate human-planner interaction. While natural language processing systems have been developed over at least the past twenty years, the advent of commodity spoken language systems and natural language processing systems (Tractica 2017) provides exciting opportunities for integration with automated PS. Augmented reality is a 'rising' technology that, when coupled with computer vision systems, can provide new, potentially disruptive methods for supporting plan execution, if not planning. There is also the potential for automated PS to help design UIs. Workflows for many different UI tools can be constructed using planning systems (St. Amant 1999) as well as other automated reasoning technologies. Historically, there have been a small number of investigations of this type.

The User Interfaces and Scheduling and Planning (UISP) workshop[1] featured two invited talks, eight presented papers (Freedman and Frank 2017), and a panel. We summarize the main content of the workshop in Sections 2 and 3. Based on the positive response to this workshop, in Section 4 we propose future directions for the community as well as an invitation to connect with other related communities.

2 The UISP 2017 Workshop

2.1 Themes in Invited Talks

User Interfaces for eXplainable Planning (XAIP) This talk focused on the need and challenges of designing UIs to enhance *transparency* and *explicability* in PS systems. While the topic of explainable AI (XAI) is mainly concerned with learning techniques (i.e., explaining neural networks), the topics of trust and transparency are also very relevant to PS. Moreover, AI Planning is potentially well-placed to

[1] http://icaps17.icaps-conference.org/workshops/UISP/

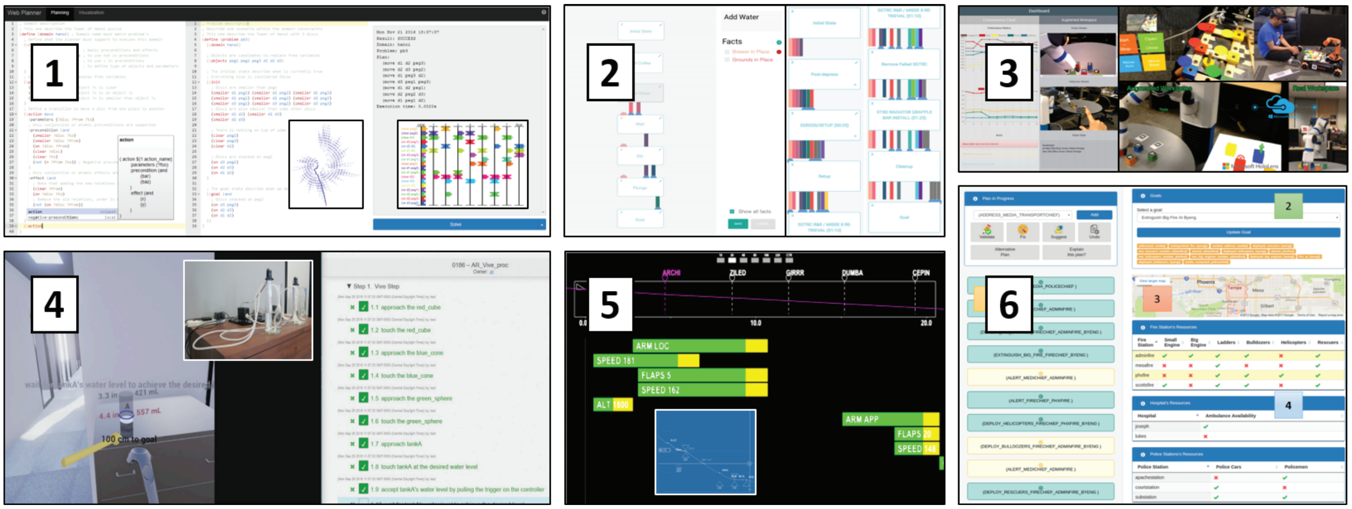


Figure 1: Snapshots of interfaces discussed in the workshop. Clockwise from top-right corner, these correspond to presentations from authors of (Magnaguagno et al. 2017; Bryce et al. 2017; Chakraborti et al. 2017; Bonasso et al. 2017; Benton et al. 2017; Sengupta et al. 2017). Salient features of these interfaces are discussed in Section 2.2 and also summarized in Table 1.

be able to address the challenges that motivate the research on XAI. *Plan Explanation* is an area of planning where the main goal is to help humans understand the produced plans. This involves the translation of the planner outputs (e.g., PDDL plans) in forms that humans can easily understand; the design of interfaces that help this understanding (e.g., spoken language dialog systems); and the description of causal and temporal relations for plan steps. Note that making sense of a plan is different from explaining why a planner made decisions, which is a key element of XAIP. However, the PS community's work in this area forms a solid basis upon which XAIP can be further developed.

This talk was based on Fox, Long, and Magazzeni (2017), which contains an overview of related work in XAIP from the planning community. Langley et al. (2017) more recently used *Explainable Agency* to refer functionalities that an autonomous agent must have in order to explain their decisions. Some of these ideas appeared earlier: Smith (2012) presented *Planning as an Iterative Process* in his AAAI invited talk, discussing the broad problem of users interacting with the planning process, which includes questions about choices made by the planner. A number of challenges for UISP in the area of XAIP were identified, including:

- UISP should help the user explore the space of alternative plans so that the user can make an informed choice;
- UISP should provide a set of plans, rather than a single plan, so that the user can choose plans according to different metrics (e.g. preferring efficiency vs. risk);
- UISP should facilitate the integration between PS technology and domain knowledge since human expertise should play a role in defining heuristics for a specific domain;
- UISP should allow the user to accept only part of a plan (rather than accepting of rejecting it as a whole);
- UISP should allow the user to add new (high-priority) goals and modify the planning model at execution time;

As noted in the Introduction, in the last few years, planners have become more powerful. PS is used in new (critical) domains (e.g., mining, energy, air/urban traffic control, etc.) that require more complex solutions (e.g., continuous non-linear models, differential equations, fluid dynamics, etc.). Prior work in explaining plans should be revisited and extended to handle these new complex scenarios.

'Want to Field Your PS System? Suck it Up!' (Challenges) This presentation surveyed case studies from a company's experience creating customized PS solutions for clients. It is frequently the case that PS technology can be applied to solve existing problems, or one can rethink of the solution to an existing problem as a planning or scheduling problem. However, it is also important to keep the clients' specifications in mind, which may require additional changes that are typically not considered at the time of designing PS technologies.

Real world problems have often been solved in some way already, which has several implications. First and foremost, the customer or stakeholders have a preconceived notion of what the problem is with respect to activities, constraints, preferences, and methods to produce good solutions (typically, but not always, heuristics for producing a plan). More importantly, from the UI point of view, there is an existing UI and a body of knowledge about what that UI should look like. Examples include specific UI elements (icons, Gantt / PERT chart elements), color choices (often with very specific meanings), desired layouts, and so on. This combination of pre-existing knowledge, process, and UI design often constrains the use of PS technology. Examples include specific knowledge that is hard to model or integrate with existing solvers, the inability to redisplay plans after new solutions are generated (either as a result of replanning or top-K plans), and the inability to display certain forms of planner output (e.g. explanations).

2.2 Themes in Presentations

PRIDE-AVR is an integration of the PRIDE (Izygon, Kortenkamp, and Molin 2008) system – which helps author, model and execute procedures that NASA flight controllers and astronauts use to manage plans – with mixed reality technologies. The system (Bonasso et al. 2017), shown in Figure 1(4), is demonstrated on three use-cases: (i) an augmented reality browser; (ii) a virtual/hybrid reality demonstration; and (iii) an on-board graphics-based system used to train astronauts for extravehicular activities.

CRADLE is a plan recognition algorithm (Mirsky, Gal, and Tolpin 2017) that analyzes users' interactions with the interface of a financial services company. The algorithm is used to decrease the amount of information that an analyst needs to consume in order to flag abnormalities and other patterns from among the various traces of a user's interactions with the financial system.

WEB PLANNER is a cloud-based planning tool that provides code editing and (search) state-space visualization capabilities. The tool (Magnaguagno et al. 2017) consists of three main interface components, shown in Figure 1(1): (i) a text-based domain and problem editor as well as a plan visualization in text; (ii) various tree-based visualizations of the search-space; and (iii) a *Dovetail* visualization, which tracks the progress of ground predicates through the state-space from the initial to goal states.

Conductor combines the plan synthesis problem with domain modeling. It uses a *"visualization metaphor derived from metro maps to display facts as transit routes and step preconditions as stations"* (Bryce et al. 2017) as shown in Figure 1(2). Insets show the visualization of these fact routes in a toy planning domain (left) and in a NASA Extravehicular Activity procedure (right).

CHAP-E (Benton et al. 2017) aims to improve aircraft pilots' situational awareness and decision making. It uses hierarchical plan representations (Figure 1(5)) and causal links (inset) to provide *"guidance toward executing procedures based on the aircraft and automations state and assists through both nominal and off-nominal flight situations."*

RADAR is a plan authoring tool (Sengupta et al. 2017) that explores the different roles of an automated planner in the deliberative process of a human planner in the loop, beyond just plan synthesis. It is the first-of-its-kind paper to explore the scope of decision support across the full spectrum of the automation hierarchy (Parasuraman and Riley 1997), especially as it relates to the role or "personality" of the automated planning assistant. Use cases are provided in a mock emergency response scenario as seen in Figure 1(6).

Æffective introduces augmented reality as an alternative vocabulary of communication in proximal operation of robots for projection of intentions and real-time feedback for replanning during a plan's execution (Figure 1(3), bottom right inset). The system (Chakraborti et al. 2017) also uses electroencephalographic signals (Figure 1(3), top right inset) to close the communication loop for preference learning and plan monitoring. A centralized dashboard (Figure 1(3), left inset) visualizes the shared brain of the agents (humans and robots) in a semi-autonomous workspace.

Complexity Metrics denote the complexity of various workflows (plans, schedules) with an eye towards collaborative, planner-assisted settings. The work's (Talamadupula, Srivastava, and Kephart 2017) main motivation is to highlight existing metrics for human comprehensibility of plans and schedules, devise a framework for evaluating existing workflows according to such metrics, and to motivate the planning community to incorporate some of these metrics into the plan synthesis process.

3 Challenges for UISP Research: Panel Discussion

The workshop included a panel discussion with representatives from academia and industry who have built a variety of PS systems, both with and without UIs. A summary of some key issues discussed by the panelists follows.

The PS research community is primarily focused on designing and evaluating algorithms to solve well-formed problems, ranging from scheduling and temporal reasoning to generating optimal policies to manage systems in the presence of uncertainty, and many problems in between. Rarely does our community build UIs for our systems, and when we do, it is typically for our own consumption (modeling interfaces, search space visualization, and so on). More crucially, the PS community is not the typical application customer, and therefore neither 'owns' nor understands the desired UI that a customer wants. In the words of one panelist, "The user will sense and perceive your planning and scheduling system entirely through its user interface." This under-appreciation of customer and particularly UI needs must be addressed to broaden the use of PS technology.

The PS community should also recognize that many applications can benefit from only a subset of existing PS algorithms. To be successful, PS algorithms must solve the customers' problem effectively; we may not need the full features of an AI planner to succeed. A related challenge is to approach problems without unnecessarily resorting to the language of AI planning, which, though formal and precise, is often hard to understand. For instance, the machine learning community claims to 'make smart decisions from data'. What is the analogous way of describing what PS technology can do for the customer?

Integrating PS algorithms with user-friendly UIs requires 'bridging the gap' between the customers' implicit and explicit needs, as well as the capabilities of the algorithms that can be brought to bear to solve the problem. An ideal team consists of the PS algorithm developers and system engineers, human factors or user experience designers who can represent the customer and oversee usability testing, and the project manager who oversees the team and manages project costs and the schedule. The mix of skills on the team ensures coverage of all the key elements for a successful project, but requires significant interaction and integration among team

Paper ↓ / Feature →	GUI	NL	MR	BCI	BE	Synthesis	Execution	Modeling	Visualization	Mixed-Initiative
PRIDE-AVR	✓	✗	✓	✗	✗	✗	✓	✓	✓	✗
CRADLE	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗
WEB PLANNER	✓	✗	✗	✗	✗	✓	✗	✓	✓	✗
Conductor	✓	✗	✗	✗	✗	✓	✗	✓	✓	✗
CHAP-E	✓	✗	✗	✗	✗	✓	✓	✗	✓	✗
RADAR	✓	✗	✗	✗	✗	✓	✗	✗	✓	✓
Æffective	✗	✗	✓	✓	✗	✗	✓	✓	✓	✗
Complexity Metrics	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗

Table 1: Features of PS interfaces presented at the workshop. (GUI = Graphical User Interface; NL = Natural Language; MR = Mixed Reality; BCI = Brain-Computer Interface; BE = Backend support for planner; Synthesis = plan generation; Execution = plan execution; Modeling = learning and authoring of planning models; Visualization = visualization of planning and execution; Mixed-Initiative = human involved in the plan generation process)

members. User experience and human factors must understand the power and capability of PS algorithms, and PS algorithm developers must recognize limitations on the solution due to the customer needs, ability to formalize the problem, and limitations imposed by the UI design.

Design iterations are critical to project success. Uncertainty about good design and capability is reduced by iteration; customers take ownership of the project as they provide feedback, and iteration can lead to the introduction of more powerful PS algorithms as users begin to appreciate what they offer. Model-based planning should be very well suited to design iteration, since models are declarative and therefore easily changed. In order to take full advantage of this, however, integration with the UI must be equally easy. One challenge of achieving integration is that most PS systems do not produce a 'standard' output format. Defining a standard output that can be easily integrated with UIs would reduce integration challenges. Many applications have preexisting UIs; thus merely ensuring a PS output standard will solve only part of the problem. Despite these limitations, an interesting challenge for the PS community is to assess existing applications and their associated UIs while considering some systems engineering questions: is there a set of 'canonical' UIs that cover a large number of applications? Can the community define a set of PS output standards (e.g. for plan generation, replanning, plan recognition, plan explanations, etc.) that cover these applications?

Finally, while it is unreasonable to expect the entire PS community to actively work on UIs, there was discussion about creating some competitions or design challenges to stimulate interest in this area. Such a competition would differ from the International Planning Competition (IPC) and Knowledge Engineering for PS (KEPS) challenges — it would focus solely on designing UIs for PS systems. While it is tempting to say that the underlying algorithms can be separated from plan displays, some amount of explanation will be required when replanning is performed (and it will be). Ultimately, deep algorithm design decisions may need to be exposed as part of the explanation.

4 Future Directions

Natural language techniques were conspicuous by their absence. Interactions in this space are especially useful while communicating with non-experts in daily life. Recent work looked at verbalization of plans and intentions in natural language (Tellex et al. 2014; Perera et al. 2016) in the context of human-robot interactions. This is an area for future growth in UISP. Perhaps the applications featured in the workshop were geared towards more structured settings with experts in the loop, where more efficient interfaces can be engineered. On the other hand, mixed reality is rapidly emerging as a major player in the space of UIs for human-computer interaction. The PS community seems to have also responded to the exciting opportunities of this emerging technology, with two out of the eight presentations departing from traditional GUIs to mixed reality systems (Bonasso et al. 2017; Chakraborti et al. 2017). A second workshop will be held at the ICAPS 2018 conference. In addition to working with the PS community, it will be important to reach out and establish collaborations with sister communities such as Intelligent User Interfaces (IUI), Human-Computer Interaction (CHI), Human-Robot Interactio (HRI and Ro-Man), and Social Computing (CSCW). This will produce the ideal teams that synergize algorithm developers and designers.

5 Discussion

The recently established UISP research community aims to bridge the gap between PS and UI technologies. The first workshop both introduced current work in this area and identified related challenges that apply to the general user experience of the AI community. With the increase in interface modalities and ubiquity of AI amongst users' lives, the research and collaboration opportunities have potential for also bridging the gap between AI and people.

References

Benton, J.; Smith, D.; Kaneshige, J.; and Keely, L. 2017. CHAP-E: A plan execution assistant for pilots. In *Proceedings of the Workshop on User Interfaces and Scheduling and Planning*, UISP 2017, 1–7.

Bonasso, P.; Kortenkamp, D.; MacIntyre, B.; and Wolf, B. 2017. Alternate realities for mission operations plan execution. In *Proceedings of the Workshop on User Interfaces and Scheduling and Planning*, UISP 2017, 8–14.

Bryce, D.; Bonasso, P.; Adil, K.; Bell, S.; and Kortenkamp, D. 2017. In-situ domain modeling with fact routes. In *Proceedings* *of the Workshop on User Interfaces and Scheduling and Planning*, UISP 2017, 15–22.

Chakraborti, T.; Sreedharan, S.; Kulkarni, A.; and Kambhampati, S. 2017. Augmented workspace for human-in-the-loop plan execution. In *Proceedings of the Workshop on User Interfaces and Scheduling and Planning*, UISP 2017, 23–31.

Chi, H.-L.; Kang, S.-C.; and Wang, X. 2013. Research trends and opportunities of augmented reality applications in architecture, engineering, and construction. *Automation in Construction* 33(Supplement C):116–122. Augmented Reality in Architecture, Engineering, and Construction.

Fox, M.; Long, D.; and Magazzeni, D. 2017. Explainable planning. *CoRR* abs/1709.10256.

Freedman, R. G., and Frank, J. D., eds. 2017. *Proceedings of the First Workshop on User Interfaces and Scheduling and Planning*. AAAI.

Izygon, M.; Kortenkamp, D.; and Molin, A. 2008. A procedure integrated development environment for future spacecraft and habitats. In *Space Technology and Applications International Forum*.

Langley, P.; Meadows, B.; Sridharan, M.; and Choi, D. 2017. Explainable agency for intelligent autonomous systems. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 4762–4764.

Magnaguagno, M. C.; Pereira, R. F.; Móre, M. D.; and Meneguzzi, F. 2017. WEB PLANNER: A tool to develop classical planning domains and visualize heuristic state-space search. In *Proceedings of the Workshop on User Interfaces and Scheduling and Planning*, UISP 2017, 32–38.

Mirsky, R.; Gal, Y. K.; and Tolpin, D. 2017. Session analysis using plan recognition. In *Proceedings of the Workshop on User Interfaces and Scheduling and Planning*, UISP 2017, 39–43.

Munteanu, C.; Irani, P.; Oviatt, S.; Aylett, M.; Penn, G.; Pan, S.; Sharma, N.; Rudzicz, F.; Gomez, R.; Cowan, B.; and Nakamura, K. 2017. Designing speech, acoustic and multimodal interactions. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*, CHI EA '17, 601–608. Denver, Colorado, USA: ACM.

Parasuraman, R., and Riley, V. 1997. Humans and automation: Use, misuse, disuse, abuse. *Human Factors: The Journal of the Human Factors and Ergonomics Society*.

Perera, V.; Selvaraj, S. P.; Rosenthal, S.; and Veloso, M. 2016. Dynamic Generation and Refinement of Robot Verbalization. In *RO-MAN*.

Sengupta, S.; Chakraborti, T.; Sreedharan, S.; Vadlamudi, S. G.; and Kambhampati, S. 2017. RADAR - A proactive decision support system for human-in-the-loop planning. In *Proceedings of the Workshop on User Interfaces and Scheduling and Planning*, UISP 2017, 44–52.

Smith, D. E. 2012. Planning as an iterative process. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*

St. Amant, R. 1999. Planning and user interface affordances. In *Proceedings of the 4th International Conference on Intelligent User Interfaces*, IUI '99, 135–142. Los Angeles, California, USA: ACM.

Talamadupula, K.; Srivastava, B.; and Kephart, J. O. 2017. Workflow complexity for collaborative interactions: Where are the metrics? A challenge. In *Proceedings of the Workshop on User Interfaces and Scheduling and Planning*, UISP 2017, 53–56.

Tellex, S.; Knepper, R.; Li, A.; Rus, D.; and Roy, N. 2014. Asking for help using inverse semantics. In *RSS*.

Tractica. 2017. The virtual digital assistant market will reach $15.8 billion worldwide by 2021.

Designing for Trust with Machine Learning

Fabien Girardin, Pablo Fleurquin
BBVA Data & Analytics, Avenida de Burgos, 16D 28036 Madrid, Spain
fabien.girardin@bbvadata.com, pablo.fleurquin@bbvadata.com

Abstract

This is a proposal for a presentation on the relation between Machine Learning and design for trust at the Designing the User Experience of Artificial Intelligence symposium as part of the 2018 AAAI Spring Symposium Series in Palo Alto, CA. Trust is at the bedrock of our human social system. Historically, the financial businesses have been based on how it could trust customers, and not the other way around. Today customers request — in addition to competence, security and lending capability — honesty, legibility, transparency and other key attributes of the trust relationship with a data-driven bank. We will share our experiments and approaches that use Machine Learning techniques to tackle mistrust and foster a trustworthy relation with our customers.

Introduction

Fabien Girardin is Co-CEO at BBVA Data & Analytics, a center of excellence in financial data analysis that aims at revolutionizing the banking industry in the domains of marketing intelligence, customer advisory, risk, fraud and the automation of financial processes. With a broad spectrum of interdisciplinary skills, he guides teams in transforming algorithmic research and experiments into value propositions, services, products and experiences that are future forward.

Pablo Fleurquin is Data Scientist at BBVA Data & Analytics with extensive experience in describing, analyzing and modelling the delay dynamics of a paradigmatic socio-technical complex system such as the air-transportation system. He uses his knowledge in Complex Network Theory, Graph Analytics and Machine Learning to develop online credit card fraud analytics, risk scoring solutions and pricing strategies.

This paper reports on our investigation and experiments that explore how the specific design of Machine Learning algorithms can consolidate trust in financial services. This work aims at orienting today how people experience banking in the near future.

An Evolution of Trust

Trust is part of a social contract with both rational and emotional bonds. Trust cannot be delivered, but actions can be taken in order to enrich it. For instance, financial businesses are based on their capacity to measure risk to grant a loan or accept a transaction. Historically, quality, transparency and altruism was demanded on the side of the customer. In consequence, a bank is often perceived as a partner people need to live with, but that are prone to mislead, provoke unfair situations and take advantage of opaque processes. That situation is changing with regulators and society, in various parts of the world, demanding openness for both protection of personal data and therefore breaking bank's monopoly in measuring risk.

Nowadays the increasing amount of digital footprint of bank customers provide with a much deeper vision to measure risk and opening new means to build trust. New analytical capacities like Machine Learning allow to transform these new datasets into personalized experiences, customized advisory with accurate forecasts, increased access to loans with less risk, as well as automated interactions.

Those technological opportunities also create design challenge that may drive mistrust between banks and their customers. The practice of data science must carefully resolve an increasing amount of dysfunctional solutions based on partial data or in bad quality data. Importantly, Machine Learning errors have totally different implications depending on the domain: the consequences are very different if we are recommending a financial product, a movie or helping with illness diagnose. Those solutions have the potential to erode trust and disengage customers, besides posing a risk proportional to the kind of service provided. A lot of research interest has been put recently on adversarial examples. These are subtle and unnoticeable changes to model inputs that an attacker intentional designs to cause the algorithm to make a mistake. For instance, in the facial-recognition field where the industry and government intelligence agencies have put a lot of effort, a recent paper has shown how by changing a small part of the image is enough to make you a different person in a machine's eyes (Sharif et al. 2017). Another research group has also shown that street-signs recognition algorithms for self-driving cars are also prone to adversarial examples. Subtle changes that a human will recognize can make an algorithm confuse a stop sign with a speed limit one (Papernot et al. 2017). In addition, discrimination like unfair access to societal goods is becoming pervasive and has reinforced the threat. We have to highlight that these technological threats, as opposed to adversarial examples, happen without any explicit wrongdoing in Machine Learning modelling. Two of the main reasons behind such a pervasive problem are sample size disparity and encoded human biases in data. The former is easy to grasp, basically minority groups are by definition under-represented in data sample, which leads to higher error rates on these groups. The latter, is part of the data and in most cases is indistinguishable from it. Biases come in many flavours: demographic, geographic, behavioural and temporal biases. Examples are becoming ubiquitous such as 2013 Ally Financial 98M US$ suit on auto-loan discrimination (McDonald and Rojc 2014). In this particular case, the Consumer Financial Protection Bureau's (CFPB) used an algorithm to infer a borrower's race based. Other border-line use of technology is in recidivism models such as the LSI-R in the United States (Whiteacre 2006). These solutions help the judicial system to assess the danger posed by each convict. A work by Caliskan et. al. 2017 showed how pre-existing biases and stereotypes permeate semantically derived word associations models. It is clear, though, that algorithms inherit human biases, that pervade historical data, and the situation is even worse when these are camouflaged into a black-box model.

Up to this point, we believe any data-driven organization like a bank we must be transparent and responsible through their decision-making process, being it algorithmically driven or not. Hence, they must detect and address potential problems to enrich a trustful relation with their customers. Our work in that domain explores the foundations of trust from a Machine Learning perspective with the basic attributes of fairness and transparency.

Fairness is always the result of a comparative process (Xia et al. 2004). This can be twofold; as a comparative process with a past personal situation or a comparative process with another person independently of time. For example, in the former case, we can consider a price increase in a certain product, given incomplete market information, as unfair. In this, anticipating the buyer discrepancies and the transparency of the vendor explaining why price has increase can reduce the sensation of unfairness. In the latter case, we base our fairness assumptions by comparing to others. Things are more intricate, because one must address, subjectively, how alike one is to the comparative others. If there is a price reduction in a certain product for people considered as peers, odds are that the comparison will provoke an unfair situation. A good example of it was the uproar that took place with Amazon dynamic pricing model when people realized that the model had charged some people more than others (Weisstein et al. 2013). Unfairness of the second type can be explicitly solved in the feature selection phase (Grgic-Hlaca et al. 2018) or including fairness metrics as another component of the algorithm development (our experiments 2 & 3).

In addition, transparency also known as Machine Learning interpretability is a key part of the toolset to tackle mistrust in algorithmic decision-making processes. It can be used to promote fairness of the first and second type, and moreover pervade the organizational culture with ethical responsibility. As the great 20th-century physicist Richard Feynman puts it: "if you cannot explain something in simple terms, you don't understand it". This maxima that is so accepted in the hard sciences, it is not that extended in Data Science. It implies a bidirectional association between explainability and understandability, which ultimately oppose transparency against blackbox-ness. It should be noted though, that black-box algorithms are not exclusively those of a non-linear nature; high dimensional and heavily tuned Generalized Linear Models can be also vastly opaque (Lipton 2016). Fortunately, interpretability frameworks clear the way to take-apart the machine and explain its pieces (our experiment 1) (Ribeiro et al. 2016; Lakkaraju et al. 2017).

An Evolution of Automation

Automation in the banking industry has come a long way since the 1970s with innovations like the Automated Teller Machine (ATM) and the Electronic Fund Transfer at Point of Sale (EFTPOS) (Consoli 2008). Automation is in the DNA of such an information driven industry. In the last years with the advent of cheap distributed databases, cloud services and computational power automation pivoted to enrich decisions algorithmically by incorporating vast and varied new data sources. Nowadays, many banks follow a digital agenda focusing on sales automation. By doing so, personalized offers reach customers at the right moment, and, in addition, automating servicing 'Do it Yourself' experiences allow for huge cost reductions on mature high margin products. Also, data-driven banks employ Machine Learning to perform more fine-grained assessment of risks and provides customized advisory.

According to McKinsey Global Institute Report (2016) Machine Learning is having a significant impact on retail banking, especially on improved forecasting and predictive analytics boosting a radical customer personalization approach (Henke et al. 2016). Nevertheless, the evolution of automation should come along with that of trust, but this coevolution is far from clear. According to an Accenture poll 87% of US consumers plan to use bank branches because of greater added value and in-person trustworthiness (Accenture 2016). Still, in general, the most valuable channel is online but not precisely because of trust as it is the reason behind branch channel value. Automation might move from traditional transactional interactions to a meaningful "relational" interaction. In an increasingly digital era, consumers are looking for experiences rather than merely servicing; a world where banks come to customers rather than customers go to the bank. Therefore, the interplay between automation and customer experience should come along together with trust, and this area is where we are putting our research efforts: how the design of automation together with Machine Learning can create trustworthy relationships with our customers.

Experiments on Trust and Machine Learning

We are currently conducting experiments that aim at understanding techniques to design for trust with Machine Learning

- Experiment 1 is about interpretability and trust in credit risk scoring: Algorithmic transparency is openness about the purpose, structure and underlying actions of the algorithms used to search for, process and decision making. This experiment explores one way of making a black-box algorithm transparent using LIME (Ribeiro et al. 2016) as an interpretability framework. By implementing this framework we can answer customer questions such as: why I have been rejected? Not only for the customer but also for the financial regulator which opens the possibility to use more sophisticated non-linear models. As well as helping risk analysts on the model development process.
- Experiment 2 is about learning to bid in real time using a fair strategy: An approach on dynamic pricing that uses Reinforcement Learning (RL) (Sutton and Barto 1998) to keep a balance between revenue and fairness. This work helps maximize revenues while taking into account fairness and equity that prevent a negative customer perception of unfair price differences that can destroy a trustful relation. We demonstrate that RL provides two main features supporting fairness in dynamic pricing: on the one hand it is able to learn from recent experience adapting the prices policy to complex market dynamics; on the other hand RL can include a trade off between short and long-term objectives, integrating fairness into the model's core. Specifically Q-learning is used to provide a simple way for agents to learn sequentially by trial and error (Watkins and Dayan 1992). In the context of our experiment it is used to, for each action performed by an agent, modify the state of the environment (related to fairness) while providing a reward (the price bid).
- Experiment 3 explores a fair approach on Recommender Systems (RS): While RS aim to provide an appealing list of items to users, most algorithms suffer from a bias in the recommendation towards popular items. As a consequence, the recommended list often goes away from the true interest of users. On the other hand, less popular, long-tail items are desirable for recommendations because of their novel and diverse character. In this experiment, we explore the concept of fairness in recommender systems, so that all items have the same chance to be presented to users. Two techniques that allow keeping a balance between popular and niche products in the recommendation are introduced. A new loss function that it is explicitly designed to deal with missing information, forbids a predicted zero preference to unseen products. This makes every product available in the recommendation. Second, a popularity-scaling factor is included in the loss function distributing the recommendation itself in a better way.

Conclusions

Trust is a complex term with multiple dimensions investigated in psychology, sociology, economics, information systems and even philosophy. From a Machine Learning perspective, we realize to only grasp the tip of the iceberg. With the new wave of Machine Learning solutions, value is created with an accumulation of touch points that feed algorithms with behavioural data. Technology can provide attributes to build trust like competence, quality, simplicity, and convenience. We have seen that Machine Learning technique can help contribute to further experiences of trust like transparency and fairness. We believe that trust is built through the intensifying relations, feedback loops, virtuous cycles, 'data network effects', and the capacity to understand and react on customer's intentions, emotions, and behaviours.

We believe that models are not sanitized abstractions of reality; on the contrary, explicitly or not, they are being created with our biases and unfair judgments. These must not be seen solely as profit seeking machines, because the choices they made in the end are fundamentally moral.

In addition, we are exploring ways to include fairness and transparency as central elements of model development, that eventually will foster a trustful relation with bank customers. We have learned one way of making opaque algorithms transparent positioning us one step ahead of the new regulatory demand which comes into force next year under the European Union General Data Protection Regulation (GDPR). Using model interpretability, we can fulfil the regulatory "right to explanation" and give feedback to customers on the decisions that affect them, as well as help data scientists and analysts on the process of training and assessing models. Regarding fairness, we are gathering empirical evidence that Reinforcement Learning is a model capable of learning revenue maximization while providing a more egalitarian dynamic pricing strategy between groups of customers. Concerning recommender systems, we developed a way of effectively dealing with the extended bias in recommendation towards popular items. Avoiding this bias means new responsible ways for the banking industry to increase its sales and profits by potentially selling in a vast and unexplored market.

References

Accenture Consulting. 2016. North America Consumer Digital Banking Survey. Banking on Value: rewards, robo-advice and relevance, Technical Report, Accenture.

Caliskan, A.; Bryson, J.J.; and Narayanan, A. 2017. Semantics Derived Automatically from Language Corpora Contain Human-like Biases. *Science*, 356(6334), pp.183-186.

Consoli, D. 2008. Systems of Innovation and Industry Evolution: The case of Retail Banking in the UK. *Industry and Innovation* 15(6), pp.579-600.

Grgic-Hlaca, N., Zafar, M.B., Gummadi, K.P. and Weller, A., 2018. Beyond Distributive Fairness in Algorithmic Decision Making: Feature Selection for Procedurally Fair Learning. Max Planck Institut Informatik

Henke, N.; Bughin, J.; Chui, M.; Manyika, J.; Saleh, T.; Wiseman, B.; and Sethupathy, G. 2016. The Age of Analytics: Competing in a Data-driven World, Technical Report, McKinsey Global Institute.

Lakkaraju, H.; Kamar, E.; Caruana, R.; and Leskovec, J., 2017. Interpretable & Explorable Approximations of Black Box Models. *arXiv preprint arXiv:1707.01154.*

Lipton, Z.C. 2016. The Mythos of Model Interpretability. *arXiv preprint arXiv:1606.03490.*

McDonald, K.M.; and Rojc, K.J. 2014. Automotive Finance Regulation: Warning Lights Flashing. *Bus. Law.,* 70, p.617.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z.B.; and Swami, A. 2017. Practical Black-box Attacks Against Machine Learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security (pp. 506-519). ACM.

Ribeiro, M.T.; Singh, S.; and Guestrin, C. 2016, August. Why should I Trust You?: Explaining the Predictions of any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135-1144). ACM.

Sharif, M.; Bhagavatula, S.; Bauer, L.; and Reiter, M.K. 2017. Adversarial Generative Nets: Neural Network Attacks on State-of-the-Art Face Recognition. *arXiv preprint arXiv:1801.00349.*

Sutton, R.S.; and Barto, A.G. 1998. *Reinforcement learning: An introduction* (Vol. 1, No. 1). Cambridge: MIT press.

Watkins, C.J.; and Dayan, P. 1992. Q-learning. *Machine learning* 8(3-4), pp.279-292.

Weisstein, F.L.; Monroe, K.B.; and Kukar-Kinney, M. 2013. Effects of Price Framing on Consumers' Perceptions of Online Dynamic Pricing Practices. *Journal of the Academy of Marketing Science* 41(5), pp.501-514.

Whiteacre, K.W. 2006. Testing the Level of Service Inventory–Revised (LSI-R) for racial/ethnic bias. *Criminal Justice Policy Review* 17(3), pp.330-342.

Xia, L.; Monroe, K.B.; and Cox, J.L. 2004. The price is unfair! A Conceptual Framework of Price Fairness Perceptions. *Journal of marketing* 68(4), pp.1-15.

Revealing Actionable Simplicity in Data

Nicholas Gisolfi
ngisolfi@cs.cmu.edu
Carnegie Mellon University

Artur Dubrawski
awd@cs.cmu.edu
Carnegie Mellon University

Abstract

We present a methodology where we identify simple structure in data, if such structure exists, and present it to end-users, enabling them to interact with data or manipulate a machine learning model. We share our bounding box algorithm which distills complex information into a small set of range rules which yield naturally intuitive visualizations. We demonstrate a few cases where simple, actionable descriptions lead to quantitative improvements in an AI pipeline.

Introduction

Big, complex data and sophisticated learning models are black-boxes to end-users, who often are not data scientists. This, however, does not absolve these end-users from responsibility for the decisions made upon recommendations from a trained Artificial Intelligence (AI) model in domains where mistakes are costly. An interface which bridges the gap between AI and human understanding is necessary to make big data and machine learning more accessible. Our aim is to identify simple structures in data and reveal interpretable information to inform the end-user's decision making, improving their comprehension of the data and model.

We define desired properties of *simple structure* which form a natural user experience. Humans are adept at reasoning in low (1, 2, and 3) dimensional spaces, making this a pragmatic criterion for maintaining simplicity. Furthermore, the low-dimensional description must involve input features with intuitive physical relevance. Input features often undergo transformations which facilitate AI processing but preclude human reasoning. It follows that simple structure should exhibit both of these desiderata - defined in the native feature space and limited to only a few features.

Any learning model with an emphasis on explainability can be used to find patterns that fit this description (Fiterau and Dubrawski 2013). We chose to use a bounding box finding algorithm to illustrate the impact of actionable simplicity. Bounding boxes are primed for success because each box is defined in a two-dimensional subspace. We force the algorithm to operate in the native feature space so it selects intuitive features.

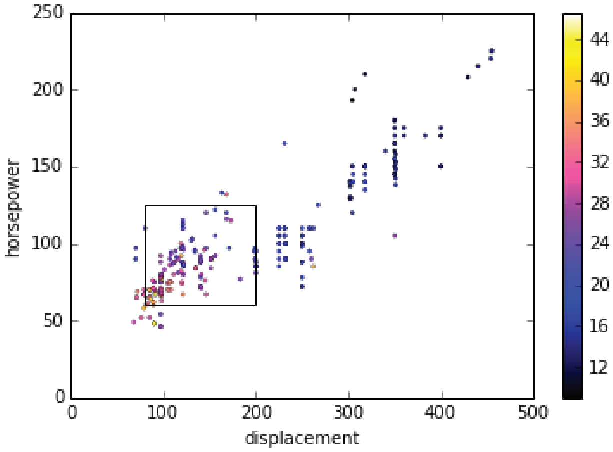


Figure 1: Example of a bounding box in a regression task.

An example of a bounding box found in a regression task on a popular benchmark dataset (”mpg” (Lichman 2013)) is shown in Fig. 1. The color labels correspond to mileage per gallon, and this box zones in on a high mean cluster. This box captures simple structure showing that vehicles with low engine displacement and low horsepower get better mileage per gallon of gasoline. Visually communicating simple structure with end-users makes it easy to interpret the meaning of the model. We will show how information like this can inform end-users to make smart decisions to improve the performance of their data and predictive models.

The Bounding Box Algorithm

Our bounding box algorithm performs a combinatorially exhaustive search across all axis-aligned, two-dimensional projections of the input feature space, looking for range rules that capture salient patterns which satisfy customizable objective criteria. Functionally, we aim to capture in these boxes statistically distinct data distributions, if they exist.

A box can be fit in a classification or regression setting, as seen in Figure 1 or 2. The resulting decision boundary is the perimeter of a box, and the box only fits the data that falls within its ranges. Each box cannot draw any conclusions for data outside its purview.

The complexity of the box finding algorithm scales

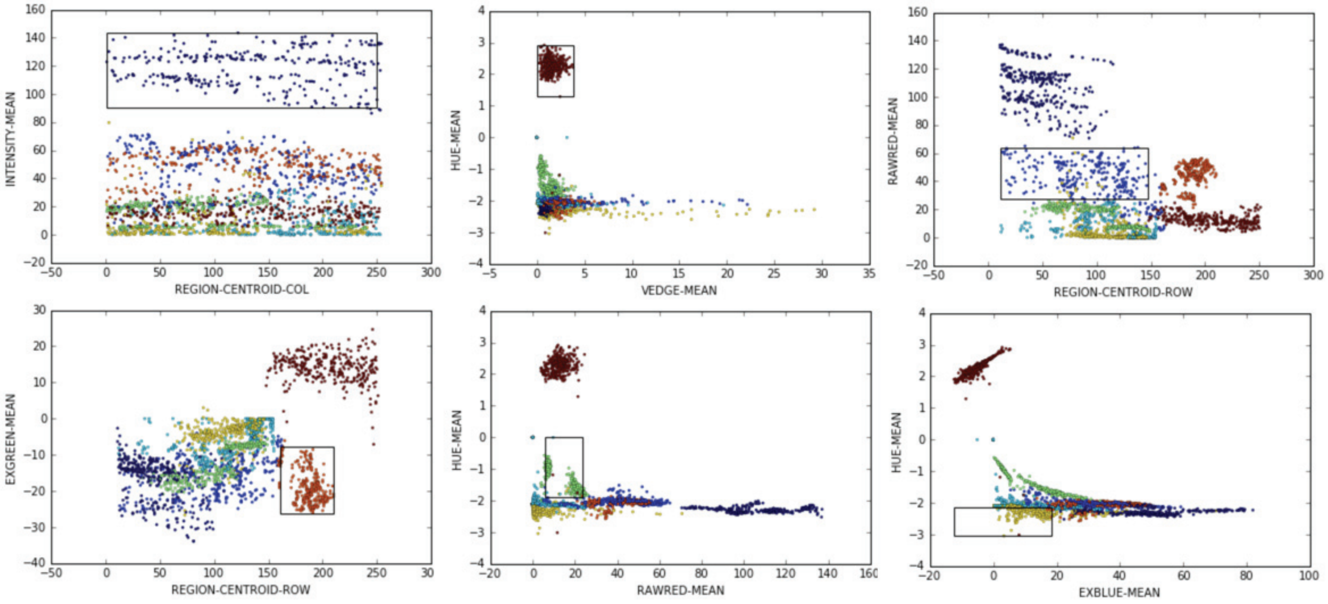


Figure 2: Sample bounding-box decision list.

with the number of features in the data. We may achieve speedups, in part, by binning individual samples to an imposed grid which minimizes the number of calls to the data structure which identifies optimal range rules. This allows us to find good boxes very quickly, or optimal boxes if given more time. Utilizing a binary search tree helps return 2D range rules in $O(n^2 \log(n))$ time, rather than $O(n^4)$ time that would be required in a naive approach. By adjusting parameters, this algorithm is capable of running on most data sets of arbitrary size in minutes, if not seconds.

In classification tasks, parameters for the algorithm include a lower bound on both purity and support size of data in the box. Purity sets a homogeneity constraint for samples inside a box. Support size sets the minimum number of samples that must reside inside a box. Variance can be used instead of purity as a box consistency metric for regression tasks. All boxes identified in a data set that meet the minimum purity and support constraints are returned for any post-processing or visualization.

Next, we share a few use cases, however the list is not comprehensive. It is possible to introduce bounding boxes in any AI pipeline to enhance the end-user experience.

Simple Structure in Big Data

Interacting with big data does not always need to be complicated. In fact, for any given set of data, there are usually some samples which are easier to classify than others. This led us to thinking that easy data may exhibit some discriminative, low-dimensional structure which can be leveraged in a classification task. If found to be true for a given set of data, this means some data may not require a complicated, comprehensive description for confident model predictions. If a large portion of a data set can be described very simply, this would make the task of understanding trends in big data much less daunting.

Table 1: Properties of datasets. All are randomly split 70/30 for train and holdout sets.

DATA SET	SAMPLES	FEATURES	CLASSES
BASEBALL	1055	16	3
PROTEIN	842	71	8
STATLOG	2100	19	7
SPECTRAL	428	101	10
THYROID	2103	28	2

To increase the descriptive power of bounding boxes for this task of identifying easy data, we form a decision list, as seen in Figure 2. The figure shows that different classes of data cluster in different subspaces, allowing us to change perspective to find multiple simple explanations. In order to produce this list, we find a single box, remove data inside from future consideration, and repeat until no boxes meet the support and purity constraints. This chains together multiple box descriptions so we have a method for combining the simple structures we find in data. The list is roughly organized by descending support size - the largest boxes are usually found first and subsequent boxes gradually describe smaller patterns left behind.

Bounding boxes provide a simple description to all data inside, therefore any sample that is captured by a box is considered *easy*. If a data sample does not fall within the bounds of any box in the decision list, then that sample is considered *hard*. Our goal is to find out how many samples in a given data set are easy, and whether the easy data exhibit discriminative simple structure.

For this experiment, we consider five publicly available datasets for classification tasks. Table 1 contains details about each data set. Running this algorithm is sometimes more art than science - instead of hand tuning input pa-

Table 2: Training data coverage, holdout data accuracy, and decision list length for multiple data sets

DATA SET	% COVERAGE	% ACCURACY	NUMBER
BASEBALL	57.4 ± 3.4	99.6 ± 0.2	1.6 ±0.5
PROTEIN	16.2± 4.4	100.0±0.0	2.8 ± 1.2
SPECTRAL	30.4 ± 7.4	99.1 ± 1.8	1.2 ± 0.4
STATLOG	38.2±11.0	100.0 ± 0.0	3.8±1.7
THYROID	54.2 ± 7.8	99.4 ± 0.2	4.6 ± 3.6
AVERAGE	39.2% ± 15.3	99.6% ± 0.3	2.8 ± 1.3

rameters based on characteristics of each dataset, we consistently set purity high and support low. High purity is the most restrictive - we will only search for boxes which are completely homogenous inside. Setting support low (minimum 5 samples in a box) makes purity the only true filter for box candidates. Running the algorithm in this way provides a lower-bound on the amount of data that we will be able to tag as easy data. By relaxing the purity constraint and hand tuning the support criterion to a specific data set, we would be guaranteed to capture more easy data.

Table 2 shows the coverage of the resulting decision lists in training data, the prediction accuracy on holdout data, and the number of boxes that comprise the decision list. The coverage of the boxes on training data is the most important detail for describing how much data is easy. Over half of the Baseball and Thyroid data sets could be captured by bounding boxes, as could an average of 39.2% of data over the five sets. The variability in coverage corresponds to the separability of classes in the data. The lowest coverage was obtained for the protein dataset, which had 8 unique class labels. The spectral set has more classes, but a larger imbalance, which makes one class more prevalent than all others.

Stating that just under 40% of data between these sets is easy does not mean much if we happen to be severely overfitting the training data. The second column shows prediction accuracies only for holdout data that falls within a box. Any holdout data that is not captured by a box in the list does not detract from the accuracy. One reason the accuracies are so high is because we demanded that all identified boxes must contain samples with homogenous class labels. The patterns captured by the boxes in training data generalize quite well to holdout data.

Lastly, the number of boxes is important for demonstrating the visibility of these simple structures in data. The average length of the decision list was less than three boxes, meaning that approximately forty percent of the data can be described with just three 2D range rules. As the wide confidence bounds indicate, the results are highly variable for different data sets, which is why it is important to tune the bounding box algorithm for each data set. For example, hand-tuning *purity=.9* and *support=50* for a single experiment with Statlog image segmentation data yields 77.3% training data coverage with 12 boxes which achieve 96% accuracy on holdout data (see Figure 2 for the first half of the list).

We can use this bounding box decision list in conjunction with a more powerful multivariate model to fall back on when the boxes do not provide output. Such a staged learning model offers comprehensibility and accuracy. Thus, the complexity of the prediction for a particular query is only as complicated as it needs to be. Tuning this staged model involves adjusting the length of the box decision list. If it turns out that a multivariate model achieves the highest accuracy without the aid of bounding boxes, then the data has no simple structure useful for this task. Otherwise, simple structure can be leveraged to make a positive impact.

Building Smarter Sets of Data

Consider a radiation threat detection system. Vehicles are scanned, as they cross international borders, for dangerous radiological signatures varying from improperly contained medical isotopes to nuclear weapons. Robust systems must be trained and validated on both benign and threat data, however, this is difficult because data corresponding to true threats either is rarely collected from the field, or is classified. A common solution to this type of problem is to reserve all empirically collected threat data for the validation set, and carefully synthesize simulated threat data to use for training.

Nuclear physicists are responsible for modeling complex physical processes which often requires generating partially synthetic, featurized data. Engineering a high-fidelity simulation over a large number of complex features is incredibly difficult and prone to omissions. Multiple small errors or oversights can lead to significant distribution drifts. Traditional machine learning assumptions, such as data being identically distributed between training and validation sets, may easily and unknowingly be violated as a result. This may negate generalization guarantees of any learning model, and this flaw may only be discovered after costly deployment procedures. Thus, it is critical to ensure that synthetic data is the most faithful representation of empirically collected data as possible.

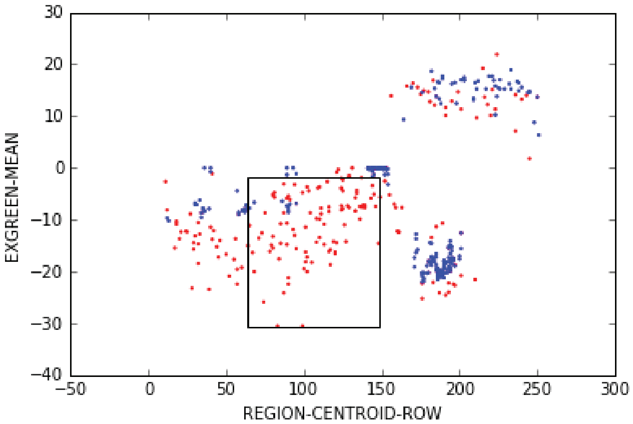


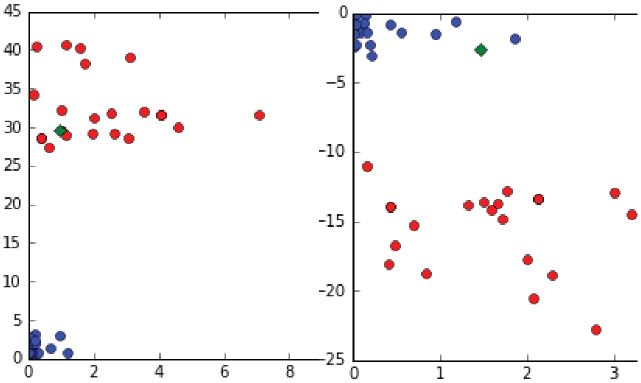
Figure 3: Significant gap illustrated between training (synthetic, blue) and validation (empirical, red) data

In order to find out how this synthesized data differs from data in the field, we define a binary classification task between training and validation data. This reveals subspaces where the distributions of select variables are significantly different between the train and validation data. Figure 3, shows a proxy example to illustrate the difference between synthetic and empirical data. The box highlights a definite mismatch in the distribution of x-axis and y-axis variables in the largest cluster. This information can guide the simulation process to make adjustments to address the gap, yet leave the other matching distributions unchanged.

Approximating the high dimensional divergence between the classes as a collection of actionable range rules allowed nuclear physicists to tackle one problem at a time. Through an iterative process, these 2D axis-aligned range rules offer a simple action item for data engineers to adjudicate. We have found that addressing shortcomings of the data in this manner improves the prediction accuracy of an existing AI pipeline without needing to make changes to the learning model itself. Generating smarter data accomplishes the same task as engineering a smarter learning model.

Contradictory Pattern Detection

A common occurrence in ensemble learning methods involves two or more models giving a single query different predictions. A simple way to resolve the contradiction would be to pick a classification that matches a majority of models in the ensemble. However, this may not always be the best strategy. Disagreements between models may signal an issue with the featurization of the data rather than just expected variability due to bootstrapping. Or perhaps the bootstrapping procedure generates poor learning models which could be discarded and replaced by changing tuning parameters.



(a) Green is similar to red

(b) Green is similar to blue

Figure 4: In different subspaces, the green diamond looks like it may belong to either the red or blue class. This is an example of a contradiction between the two models.

We search for simple explanations for disagreements between multiple models using the bounding box algorithm. In this demonstration, we train a random forest model containing an even number of decision trees. We allow the random forest to resolve internal contradictions if plurality exists between models, however, in the case of a tie, we try to find a simple explanation for why this large-scale disagreement manifests. Both the conflicting sample and the respecting leaf nodes of interest are identified. A binary classification task is defined between the samples that each tree identifies as most similar to the query in conflict.

In Figure 4, red samples belong to the first decision tree (T1) and blue samples belong to the second decision tree (T2). The conflicting query is marked with a green diamond. Out of the many projections identified, two are hand selected which most clearly show the cause of the disagreement. Based on the Figure 4a, T1 seems justified to classify the query as red. Similarly, Figure 4b suggests T2 has reason to classify the query as blue. Instead of just presenting a tie as the result of a vote, we are able to present a physical manifestation for the cause of the tie. This not only allows an end-user to collapse the tie with their own adjudication, but they may also alter the learning model to prevent the tie from repeating in a similar situation. In any case, presenting the end-user with a visual narrative for a contradiction will facilitate a correction making the data or model more robust.

Conclusion

By leveraging simple structures in data we craft a visual interface for end-users which enables them to interact with data and learning models. Our method for finding simple structures is fast, intuitive, and model-agnostic, allowing it to be used in many different ways when explainability is desired. Meaningful patterns that were previously buried under massive amounts of data are now able to be identified and guide an action plan for improving an AI pipeline by building smarter data, choosing appropriate models, or resolving contradictions between multiple models.

We have shown a few use cases where simple structure makes a big impact. Big data often has hidden simplicity that can be used to present information of interest to end-users. Domain experts can iteratively address shortcomings in synthetic data to create a more effective set of data. Detecting contradictory patterns allows end-users to resolve disagreements in ensemble methods and prevent identical confusion from happening in the future. We continue to study the utility of actionable simplicity across a variety of domains and applications where AI is meant to be an extension of human reasoning, not an autonomous substitute.

Acknowledgments

This work has been partially supported by Domestic Nuclear Detection Office under award 2014-DN-077-ARI087-01, by National Science Foundation award 1320347, and by DARPA under awards FA8750-12-2-0324 and FA8750-17-2-0130. Data sets in this paper are all publicly available from the UCI machine learning repository.(Lichman 2013)

References

Fiterau, M., and Dubrawski, A. 2013. Informative projection recovery for classification, clustering and regression. In *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, volume 1, 15–20. IEEE.

Lichman, M. 2013. UCI machine learning repository.

Artificial Digitality

Kuldeep Gohel
Parsons School of Design
gohek510@newschool.edu

Abstract

This paper discusses the technical process and artistic intent of a concept music album co-authored by a human and Artificial Intelligence. The project finishes with an album of several compositions. The album begins with a composition that is generated by the author alone, a self taught musician and a technologist. The compositions that follow are co authored by an open source neural network and the author. The neural network is trained by the author, who has turned his compositions into mathematical data which can be fed to the neural network. The album ends with a composition that is completely generated by the trained neural network. The goal of the project is to express the rise of Artificial Intelligence in a musical way and speculate on the future of Artificial Intelligence. The author uses music, mathematics, and the emerging machine learning field to create a musical story. It aims to question about the future where automation takes over the human labor in various fields including creative areas like music production and art making.

Context

The project is driven by two forces: the author's love for music composition and artificial intelligence. The author is a self-taught musician who has indulged himself in the process of music composition for a long time. For the past 5 years, the author has been constantly working on creating music and has recently ventured into using AI as a tool for creating music.

The author in these five years of music learning has been involved in various other studies related to emerging technologies and several art projects. In this time, he heard about Artificial Intelligence and fell in love with the idea that a machine can replicate him and help him to produce music that he is unable to give time to. He considers machine learning as a tool to replicate his brain and generate a body—in the computer—that he can share his soul with.

This thinking has led him to work on researching about Artificial Intelligence and machine learning that he supposes will be taking over the world very soon. The author's constant effort to achieve a system that can replicate his style of music production has finally led him to a neural network and a mathematical system that clones his musical passion.

The author aims to generate an album where machine learning is used to make music and the compositions are used as a medium to express the rise of Artificial Intelligence. Along with this he aims to speculate on the future of Artificial Intelligence and figure out the possible ways A.I. will assist humans in the future.

Process

The process involved an analytical approach to the art of music making. The author analyzed the process of making music, then gathered and converted the process into data; number and sequences, which can be used to generate a system that will clone his music making.

To eliminate the amount of errors, the author first limited the notes on the piano and the scale (for each composition) that he and the neural network can work with. C Natural Minor, C# Melodic Minor, G# Hungarian Gypsy Scale were selected for the three compositions. Two octaves (15 keys) were selected for each composition. The author than composed the first one-minute brief piece, called "Necessity is the Mother of Invention" that describes the world before AI and computers. The composition consists of chords and melody. Chords (limiting to 3 notes played at once) had one variable which was the sequence of notes on the piano while melody had two variables; time and note sequence. The selected two octaves of the piano scales and their respective keys were numbered from 1-15 and these numbers were used to get the sequence from the author's first composition. Three kinds of sequences were extracted; the chord notes, the time between each note in melody and the note in the melody. The value of these sequence was normalized between 0-1. These sequences where then used to make data sets (input and target data) to feed to the SRN network generated in the open source software. Three neural networks were trained to give three different sequences; for chord notes, melody notes and melody time.

The trained system was finally used to make the other two compositions, where the first one was composed by the author, the second one by the neural network and the author, and the album concludes with a music piece completely generated by the trained neural network.

References

Alpaydin, E. 2016. Machine learning: the new AI. Cambridge, MA: The MIT Press.

Russell, S. J., & Norvig, P. 2016. Artificial intelligence: a modern approach. Boston: Pearson.

Perricone, J. 2007. Melody in songwriting: tools and techniques for writing hit songs. Boston, MA: Berklee Press.

Towards Natural Cognitive System Training Interactions: A Preliminary Framework

Erik Harpstead,*[1] Christopher J. MacLellan,*[2] Robert P. Marinier III,[2] Kenneth R. Koedinger[1]

[1]Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, PA, 15213
[2]Soar Technology, Inc. 3600 Green Court, Suite 600 Ann Arbor, MI 48105
* These authors contributed equally and should be considered co-first authors.
eharpste@cs.cmu.edu, chris.maclellan@soartech.com, bob.marinier@soartech.com, koedinger@cmu.edu

Abstract

Researchers have developed cognitive systems capable of human-level performance at complex tasks (e.g., Watson and AlphaGo), but constructing these systems required substantial time and expertise. To address this challenge, a new line of research has begun to coalesce around the concept of cognitive systems that users can teach rather than program. A key goal of this research is to develop **natural** approaches for end users to directly train these systems to perform new tasks. However, what makes training interactions natural remains an open research question that we begin to explore in this paper. To lay the foundation for this exploration, we review the human-computer interaction literature to identify characteristics of systems that have historically been natural for end users to interact with. Based on this review, we propose a framework for cognitive system training interactions that decomposes interaction into *patterns*, *types*, and *modalities*, all of which support the acquisition of different kinds of *knowledge*. Finally, we discuss how this framework characterizes existing research within this space and how it can guide future research.

Introduction

In recent years, there has been a growth of research and development in the area of cognitive systems, or systems capable of higher-level processing and reasoning with structured representations using techniques informed by cognitive science (Langley, 2012). For example, IBM's Watson and Google's AlphaGo systems have demonstrated that it is possible for cognitive systems to achieve human-level performance at complex tasks. However, cognitive systems still remain largely out of reach for the general public (Laird et al., 2017). A major factor contributing to this disconnect is that our daily lives are filled with a wide range of tasks across multiple domains, whereas today's state-of-the-art cognitive systems are implemented to perform specific tasks in specific domains. Extending specialized cognitive systems to support a wider range of tasks requires substantial time and expertise (e.g., the base IBM Watson system that famously beat two Jeopardy! champions required over a century of AI expert development time).

To address this challenge, cognitive systems researchers have begun exploring approaches for users to create and extend the capabilities of cognitive systems by teaching them, rather than by programming them. This emerging area of research, which has been referred to as Interactive Task Learning (Kirk & Laird, 2014; Laird et al., 2017) and Apprentice Learning (MacLellan, 2017; MacLellan, Harpstead, Patel, & Koedinger, 2016), aims to develop the computational and cognitive theory needed for building systems that support natural interactions and that possess general capabilities for learning across a wide range of domains and contexts. Similar to how research and development on computing hardware enabled the transition from corporate mainframes to personal computers, this research area aims to support the transition from monolithic cognitive systems (e.g., Watson) to personal cognitive systems (e.g., Forbus & Hinrichs, 2006).

The longer-term goal of our research program is to develop a user-centered approach for teaching cognitive systems. For the moment, we will focus on the issue of naturalness and in particular the naturalness of the training interactions these systems afford. In doing so, we draw on the human-computer interaction perspective that an understanding of interaction is central to the design and development of usable technology. In this paper, we first review commonly recognized characteristics of natural interaction from the HCI literature and propose a preliminary framework that characterizes the space of training interactions that cognitive systems could support. Ultimately, we intend this work to lay the foundation for the development of personal cognitive systems that users can naturally teach.

What Makes an Interaction Natural?

In order to create an initial framework for natural training interactions, we must first contend with what it means for an interaction to be natural. While it is common to think of gesture and speech as lending naturalness to an interaction, the prior literature highlights that an interaction is not necessarily natural by virtue of its physical modality. Norman (2010) argues that so called natural user interfaces (e.g., speech- and gesture-based) are not inherently more natural than graphical user interfaces (e.g., screen-based widgets). For example, gestural interfaces lack the affordances to let users know what gestures they support, whereas graphical user interface widgets, such as buttons, readily advertise their supported interactions. In general, this work suggests that the naturalness of a modality alone is neither necessary nor sufficient for making an overall interaction natural.

Given that naturalness does not derive from modality, then what makes interaction natural? To address this question, we reviewed the HCI literature on natural interactions and identified four common characteristics of systems that support naturalness: they (1) support the goals of the user, (2) do what the user expects, (3) allow the user to work the way they want, and (4) leverage users' experience to minimize training. In this section, we review each of these characteristics.

Supports the goals of the user. Systems supporting natural interactions should be able to support what users want to do (i.e., their goals). One temptation in developing these systems is to overemphasize ease of use at the expense of limiting what users can achieve. Myers, Hudson, and Pausch (2000) refer to this balance as the threshold and ceiling of tools. Thresholds refer to the barriers a user must overcome to use a tool, whereas the ceiling describes what the tool enables users to do. Many systems attempting to support natural interactions emphasize a low threshold, but often ignore the ceiling. For example, it is easy to interact with Siri, but it only supports built-in commands—it is unable to learn new commands. To overcome this risk, systems should be developed with end-user goals and intents in mind (e.g. the desire to teach Siri new user-defined commands), so that the developers can ensure the system does not limit users' capabilities.

Does what the user expects. A common theme in research on natural interactions is an emphasis on the expectations users have for a system (Myers, Pane, & Ko, 2004). Humans typically follow patterns, scripts, or norms when engaging in everyday interactions (Bicchieri, 2006), which make it possible for the humans involved in the interaction to know how to respond. For example, tutors generally expect that their pupils will attempt to solve problems before asking for help. Systems that aspire to naturalness should support naturally occurring patterns of interaction and be aware of users' expectations within these patterns. It is worth noting that these patterns may arise from a user's particular cultural background (e.g., what roles their culture ascribes to teachers and students) or from their personal experiences (e.g., whether they are a Mac or PC user). Additionally, systems attempting to be natural should not require users to learn new (unnatural) patterns of interaction—deviations from typical scripts make it difficult for users to know what the system will do next and how to respond accordingly.

Allows the user to work the way they want. Given that natural systems support users' goals they should also let users execute those goals the ways they prefer or expect to. A key idea from the ubiquitous computing literature is that computing systems should become invisible because they seamlessly support the ways users want to do something (Weiser & Brown, 1996). They should not impede users or force them to achieve goals in unpreferred ways. For example, a common trend is to build systems around a speech interaction paradigm, but there are many situations where speech is an unnatural form of communication. In his study of architectural designers, Schön (1982) found that sketches of designs often better supported communication and reasoning than verbal articulations. This finding suggests that systems aiming to support natural architectural design should prefer sketch-based interactions over speech.

Leverages users experience to minimize necessary training. One of the most pervasive ideas within research on natural user interfaces is the idea of instant expertise (Wigdor & Wixon, 2011), or the idea that users should not have to learn how to control a system because the modality used is one they have immediate familiarity with. In the words of Buxton (Larsen, 2010), "[*natural user interfaces*] *exploit skills that we have acquired through a lifetime of living in the world, which minimizes the cognitive load and therefore minimizes the distraction".* Common approaches within this space include voice- and text-based natural language and gestural interfaces that take advantage of users' lived experiences interacting with other people. Additionally, many users have extensive training with artificial interfaces, such as QWERTY keyboards, that may be natural for many application contexts, so it is worth noting that these artificial modes of interaction should not be discounted.

A Preliminary Framework for Cognitive System Training Interactions

In order to design cognitive systems that support natural training interactions, we require a better understanding of how these systems could hypothetically interact. In this section, we will propose a framework for cognitive system training interactions that aligns with the four characteristics noted in the previous sections. We do not intend for this

Table 1. A Framework for Designing Natural Training Interactions for Cognitive Systems

Knowledge	Patterns	Types	Modalities
• Goals • Beliefs • Concepts • Experiences • Skills • Dispositions	• Passive Learning • Operant Conditioning • Direct Instruction • Apprentice Learning • After-Action Review • Socratic Learning • Collaborative Learning	• Command • Clarify • Acknowledge • Inform • Spotlight • Annotate • Reward • Demonstrate • Request <type>	• Command-Line Interface • Control device • GUI • Sketch • API • Gesture • Speech • Text • Multi-modal

work to be complete but hope that it provides a useful language to start talking about naturalness in the context of cognitive systems and their instructional interactions with users.

The framework characterizes four dimensions of training interactions between an agent and a human. First, we assume the goal of an interaction is to change some aspect of an agent's **knowledge**. The interplay between agents and trainers follow instructional **patterns**. Within patterns, trainers engage in several **types** of interaction, and these interactions can be done through various **modalities**. Table 1 shows these four aspects of training interactions and presents examples of each.

Knowledge. The goal of any training interaction is to update the learner's knowledge. There are many types of knowledge that might be included in a cognitive system. However, within the literature, there are several generally accepted types of knowledge (Laird, Lebiere, & Rosenbloom, n.d.). For our preliminary framework, we include six such kinds of knowledge: *goals*, which fully or partially describe desirable states of the world; *beliefs*, which represent an agent's current worldview; *concepts*, which support semantic inference and enable an agent to augment its worldview with additional non-observable information; *experiences*, which organize past situations and problem-solving episodes; *skills*, which describe procedures for changing the world and updating beliefs; and *dispositions*, which specify an agent's problem-solving orientations (e.g., whether to explore or exploit). Our current focus is primarily on symbolic forms of knowledge arising from interactions with a trainer, but future extensions of the framework might also include sub-symbolic knowledge (e.g., learning probabilistic grammar knowledge for parsing English sentences or equations as in Li et al. (2015)). Further, we do not mean to imply that all cognitive systems must support all of these knowledge categories but rather that the nature of the knowledge being changed will likely dictate choices across the other dimensions of the framework.

Patterns. Within human-human instructional settings there are many naturally occurring interaction and training patterns. These patterns govern the relationship between trainer and trainee and establish the contours for how training interactions play out. Inspired by existing systems (Hinrichs & Forbus, 2014; Kirk & Laird, 2014; MacLellan et al., 2016) and instructional practice (Chi & Wylie, 2014; Koedinger, Booth, & Klahr, 2013), our framework highlights several possible patterns. At its most simple, learning could be primarily passive, with agents observing training behaviors without active input from instructors. Increasing complexity, agents can have some control over their actions and receive rewards from the environment or an instructor (operant conditioning) or instructors can explicitly coach an agent, without requiring agent decision making (direct instruction). An even more complex pattern, apprentice learning (MacLellan et al., 2016), incorporate aspects of both of these approaches—both explicit instruction and feedback on agent actions. Additionally, many other instructional patterns are possible, such as after-action review, Socratic learning (Chi & Wylie, 2014), and collaborative learning (Olsen, Belenky, Aleven, & Rummel, 2014).

Types. Within a pattern, an instructor and trainee engage in many types of interactions. For example, within the apprentice learning pattern (MacLellan et al., 2016), an instructor issues a *command*, which specifies the task for an agent to perform. If the agent does not know how to perform the task, then it might *request* a *demonstration* from the instructor, who provides one. On subsequent tasks, the agent might attempt the task (i.e., provide the instructor with a *demonstration*) and *request* feedback (i.e., a *reward*) on this attempt. Finally, the instructor provides the agent with the appropriate *reward*. Under this pattern, this process continues until the agent is correctly performing all of the tasks. Our framework also includes interaction types for supporting Direct Instruction (Hinrichs & Forbus, 2014; Kirk & Laird, 2014), which allow instructors to directly *inform* agents about the world ("TicTacToe is a two-player game"), *spotlight* agents attention on particular parts of the world ("This [pointing] is a block"), and *annotate* demonstrations ("This is the move action [demonstrate drawing of X on board]") to facilitate efficient learning. The types listed in Table 1 are drawn from existing systems as well as the literature on communicative acts (Allen, Blaylock, & Ferguson, 2002; Traum & Hinkelman, 1992). This is not meant to be an exhaustive list, but is representative of the types that commonly occur in current practice. It is important to note that when we refer to interaction types we are interested in the overall instructional act being performed and not how it is being performed. For example, orders delivered via a command line interface or spoken natural language are both instances of the *command* type.

Modalities. The different types of interactions ultimately ground out in particular modalities of interaction, with many different modalities, or potentially multiple simultaneous modalities, supporting each type. For example, command-line or graphical-user interfaces, are both capable of supporting all of the interaction types listed in Table 1. Typically, systems that claim to support natural interaction leverage modalities commonly used in human-human interaction as the primary modes of interaction. For example, the Microsoft Kinect enables gesture- and speech-based interactions. A key aspect of modalities from our perspective is that they are cast in terms of what the trainer is doing and not necessarily how an action is being detected by an agent. For example, a gesture such as waving could be detected using either visual sensing with a camera or gyroscopic sensing with a wearable device (e.g., Taylor, Quist, Lanting, Dunham, & Muench, 2017); in either case, the trainer would be using a gestural modality.

These four dimensions intentionally map to the four characteristics highlighted in the previous section. In particular, in the context of training, supporting a user's goals consists of supporting of the types of knowledge transference they are trying to achieve. Users' expectations regarding training will derive from the social instructional patterns they have experience with. Thus, in order to naturally support training interactions, it is important for system designers to be aware of the interaction patterns that users expect. Further, users will want to interact in certain ways and system designers should be aware of the different types of interactions they want to perform. Finally, for each type of interaction, system designers should leverage modalities that draw on users' prior experience.

Other Existing Frameworks

The concept of decomposing human-agent interactions using a framework is not new and multiple decompositions exist in the prior literature. For example, Laird et al. (2017) divide interactive task learning systems by the mode of communication used (natural language or demonstration) and the type of knowledge taught (goals, concepts, actions, and procedures). Our work differs in that it also emphasizes the importance of higher-level interaction patterns, such as passive learning, direct instruction, and apprentice learning. Many interactive task learning systems use a pattern similar to apprentice learning, so this dimension may have less variation within that literature. Additionally, we make a distinction between interaction types and modalities because it is possible for interactions to be communicated via different modalities, such as a demonstration (an interaction type) being communicated using sketch, speech, or a graphical user interface (different modalities).

Another related line of work is Bartneck and Forlizzi's (2004) human-robot interaction framework, which, like our framework, has categories for patterns—called norms—and modalities. However, this framework focuses on robot's social interactions with humans more generally, rather than training interactions specifically, and so does not have dimensions for the types of knowledge being taught. Additionally, we emphasize interaction types, which form an intermediate layer of abstraction between patterns and modalities. Finally, as their work emphasizes the physicality of robots, it also distinguishes systems by the form they take (e.g., abstract or anthropomorphic). However, as our work is less concerned with the physical embodiment of agents, we do not make this distinction, but it is not incompatible with our current thinking. In general, while many existing frameworks share commonalities with the one proposed here, their focus is either more general (interaction broadly) or directed toward a different kind of interaction (non-training interactions). Thus, we believe our framework combines prior ideas, but still presents a novel perspective on interaction that is better aligned with our high-level goal of building cognitive systems that are natural for end users to train.

Discussion and Future Work

In proposing this initial framework, we aim to achieve three objectives. First, we attempt to highlight what we view as a key opportunity within cognitive systems research: to better understand the space of training interaction and develop cognitive systems that are natural and efficient for users to teach and interact with. Recent research efforts, such as Rosie (Kirk & Laird, 2014), the Companion Architecture (Forbus & Hinrichs, 2006), and the Apprentice Learning Architecture (MacLellan et al., 2016), have begun exploring different combinations of patterns, types, and modalities to support training interactions with end users. Each of these systems represent particular choices across the dimensions of our framework. To reach a more complete understanding of the space of training interaction design, researchers should explore additional approaches and new combinations of approaches in order to explore the space more broadly and ultimately direct work toward designing more natural means for training cognitive systems.

Second, organizing training interactions along an orthogonal set of dimensions enables a modular approach to the challenge of building cognitive systems to support natural training interactions. Individual researchers or developers need not contend with the whole problem and can instead focus on addressing subproblems. For example, one team of researchers might investigate which patterns are best for acquiring skills knowledge, whereas another team might investigate which patterns are best for acquiring concepts. Because these decisions are orthogonal, both teams can benefit from each other's work and integrate their findings within the common structure of the framework to support the development of systems that can naturally learn both skills and concepts. Thus, the framework supports the unification of independent research efforts, even if these efforts do not explicitly describe their work within this framework.

Finally, towards the goal of actually building cognitive systems that people can naturally train, we intend our framework to provide a language for formulating scientific hypotheses about how such systems should interact with users to best achieve naturalness. Much of the existing work implicitly assumes that choosing natural approaches for only one of the components of the framework (patterns, types, or modalities) establishes the overall naturalness of a system. For example, Hinrich and Forbus (2014) emphasize the use of multiple natural modalities, such as text and sketching, whereas MacLellan et al. (2016) emphasize the use of a natural pattern. Central to our framework, however, is the hypothesis that different combinations of patterns, types, and modalities of interaction are better suited for updating different kinds of knowledge. Thus, we believe that systems that are natural for users to teach will not only support a wide range of patterns, types, and modalities, but flexibly choose the appropriate combination based on the type of knowledge being communicated, the trainer's preference, and potentially other contextual factors. There is evidence that learning in humans follows a similar logic, in that different kinds of knowledge are best taught by different forms of instruction (Koedinger, Corbett, & Perfetti, 2012). Given that an artificial intelligence need not represent a natural system, there is no inherent reason to transfer this logic (Simon, 1983). However, if we want to support humans in naturally training such systems, then it becomes important to understand these relationships and how they might impact different kinds of training. In conclusion, it is our hope that this framework will focus attention on this issue, provide a language for talking about training interactions and their naturalness, and guide future research on this exciting frontier of personal cognitive systems.

References

Allen, J., Blaylock, N., & Ferguson, G. (2002). A problem solving model for collaborative agents. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems part 2 - AAMAS '02* (p. 774). https://doi.org/10.1145/544862.544923

Bartneck, C., & Forlizzi, J. (2004). A design-centred framework for social human-robot interaction. In *Proceedings of the13th IEEE International Workshop on Robot and Human Interactive Communication - Ro-Man 2004* (pp. 591–594). https://doi.org/10.1109/ROMAN.2004.1374827

Bicchieri, C. (2006). *The grammar of society: the nature and origins of social norms*. Cambridge University Press.

Chi, M. T. H., & Wylie, R. (2014). The ICAP Framework: Linking Cognitive Engagement to Active Learning Outcomes. *Educational Psychologist*, *49*(4), 219–243. https://doi.org/10.1080/00461520.2014.965823

Forbus, K. D., & Hinrichs, T. R. (2006). Companion cognitive systems: a step toward human-level AI. *AI Magazine*, *27*(2), 83–95. https://doi.org/10.1609/aimag.v27i2.1882

Hinrichs, T. R., & Forbus, K. D. (2014). X Goes First: Teaching a Simple Game through Multimodal Interaction. *Advances in Cognitive Systems*, *3*, 31–46.

Kirk, J. R., & Laird, J. E. (2014). Interactive Task Learning for Simple Games. *Advances in Cognitive Systems*, *3*, 13–30.

Koedinger, K. R., Booth, J. L., & Klahr, D. (2013). Instructional Complexity and the Science to Constrain It. *Science*, *342*(6161), 935–937.

Koedinger, K. R., Corbett, A. T., & Perfetti, C. (2012). The Knowledge-Learning-Instruction Framework: Bridging the Science-Practice Chasm to Enhance Robust Student Learning. *Cognitive Science*, *36*(5), 757–798. https://doi.org/10.1111/j.1551-6709.2012.01245.x

Laird, J. E., Gluck, K., Anderson, J., Forbus, K. D., Jenkins, O. C., Lebiere, C., … Kirk, J. R. (2017). Interactive Task Learning. *IEEE Intelligent Systems*, *32*(4), 6–21.

Laird, J. E., Lebiere, C., & Rosenbloom, P. S. (n.d.). A Standard Model for the Mind: Toward a Common Computational Framework across Artificial Intelligence, Cognitive Science, Neuroscience, and Robotics. *AI Magazine (Special Issue on The Cognitive System Paradigm: A New Thrust to Attain Human-Level AI)*, In Press.

Langley, P. (2012). The Cognitive Systems Paradigm. *Advances in Cognitive Systems*, *1*, 3–13.

Larsen, L. (2010). CES 2010: NUI with Bill Buxton. Retrieved April 10, 2017, from https://channel9.msdn.com/Blogs/LarryLarsen/CES-2010-NUI-with-Bill-Buxton

Li, N., Matsuda, N., Cohen, W. W., & Koedinger, K. R. (2015). Integrating representation learning and skill learning in a human-like intelligent agent. *Artificial Intelligence*, *219*, 67–91. https://doi.org/10.1016/j.artint.2014.11.002

MacLellan, C. J. (2017). *Computational Models of Human Learning: Applications for Tutor Development, Behavior Prediction, and Theory Testing*. Carnegie Mellon University.

MacLellan, C. J., Harpstead, E., Patel, R., & Koedinger, K. R. (2016). The Apprentice Learner Architecture: Closing the loop between learning theory and educational data. In *Proceedings of the 9th International Conference on Educational Data Mining - EDM '16* (pp. 151–158).

Myers, B., Hudson, S. E., & Pausch, R. (2000). Past, present, and future of user interface software tools. *ACM Transactions on Computer-Human Interaction*, *7*(1), 3–28. https://doi.org/10.1145/344949.344959

Myers, B., Pane, J., & Ko, A. (2004). Natural programming languages and environments. *Communications of the ACM*, *47*(9), 47. https://doi.org/10.1145/1015864.1015888

Norman, D. a. (2010). Natural user interfaces are not natural. *Interactions*, *17*(3), 6. https://doi.org/10.1145/1744161.1744163

Olsen, J. K., Belenky, D. M., Aleven, V., & Rummel, N. (2014). Using an intelligent tutoring system to support collaborative as well as individual learning. In *Proceedings of the International Conference on Intelligent Tutoring systems - ITS 2014* (pp. 134–143). Retrieved from http://link.springer.com/chapter/10.1007/978-3-319-07221-0_16

Schön, D. A. (1982). *The Reflective Practitioner: How Professionals Think in Action*. New York, New York, USA: Basic Books.

Simon, H. A. (1983). Why Should Machines Learn? In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine Learning*. Springer Berlin Heidelberg.

Taylor, G., Quist, M., Lanting, M., Dunham, C., & Muench, P. (2017). Multi-Modal Interaction for Robotics Mules. In *Proc. SPIE 10195, Unmanned Systems Technology XIX, 101950T (5 May 2017)*. https://doi.org/http://dx.doi.org/10.1117/12.2262896

Traum, D. R., & Hinkelman, E. A. (1992). Conversation Acts in Task-Oriented Spoken Dialogue. *Computational Intelligence*, *8*(3), 575–599. https://doi.org/10.1111/j.1467-8640.1992.tb00380.x

Weiser, M., & Brown, J. S. (1996). Designing Calm Technology. *PowerGrid Journal*, *1*(1), 75–85. https://doi.org/10.1.1.135.9788

Wigdor, D., & Wixon, D. (2011). *Brave NUI World: Designing Natural User Interfaces for Touch and Gesture*. Burlington, MA: Morgan Kaufmann.

Design Methods to Investigate User Experiences of Artificial Intelligence

Karey Helms,[1] Barry Brown,[2] Magnus Sahlgren,[3] Airi Lampinen[2]

KTH Royal Institute of Technology, Stockholm, Sweden[1]
Stockholm University, DSV, Kista, Sweden[2]
RISE SICS, Kista, Sweden[3]
Corresponding author: karey@kth.se

Abstract

This paper engages with the challenges of designing 'implicit interaction', systems (or system features) in which actions are not actively guided or chosen by users but instead come from inference driven system activity. We discuss the difficulty of designing for such systems and outline three Research through Design approaches we have engaged with - first, creating a design workbook for implicit interaction, second, a workshop on designing with data that subverted the usual relationship with data, and lastly, an exploration of how a computer science notion, "leaky abstraction", could be in turn misinterpreted to imagine new system uses and activities. Together these design activities outline some inventive new ways of designing User Experiences of Artificial Intelligence.

Introduction

There has been a growing interest in technology preempting our needs, with at least the potential of systems that are contextual, anticipatory and personalized, drawing on objects and bodies embedded with sensors and actuators. While progress has been at times halting, we are no longer surprised at the idea of cars that automatically park themselves, toilet paper that preemptively replenishes stock, or virtual assistants that sensitively diagnose diseases. These smart technologies potentially offer the possibility to transform our everyday lives, catalyzing a shift from explicit interactions towards implicit interactions.

One way of characterizing these possibilities is in a change from *explicit* to *implicit* interactions (Ju and Leifer 2008). While explicit interactions demand our immediate attention for direct engagement or manipulation, implicit interactions rely on peripheral information to seamlessly behave in the background until appropriately shifted into attention. Systems like the Google Nest automatically change household temperature based on the predicted presence of household inhabitants, offering a tantalizing sense of systems that pre-empt our needs.

Yet, in reality the inevitable choreography between implicit and explicit interactions and the resulting user experiences are far from seamless, secure and sure. Automatic doors jerk and stutter, digital products and services uncannily act upon our behalf, manipulating our emotions, or curating filtered experiences without an ability to inquire or intervene. Content is hidden from us without our permission, and in extreme cases, systems take pre-emptive actions – resetting for system upgrades just before a talk, or suspending activity until impossible conditions are satisfied.

In our own recent work, we have focused on how AI and Machine Learning techniques can be used to support the choreography between these implicit and explicit user and system actions. Working in this area is challenging because while a system might pre-empt a user action, error rates - as well as unforeseen actions - can hinder utility. It is not clear that focusing simply on automating existing applications and system actions is as useful as expected – the track record of pre-emptive system actions is mixed at best.

What is perhaps needed is a design perspective on implicit systems, deploying design methods to understand and conceptualize how the developing form of AI systems might be deployed in actual systems. In our research, we are focusing on exploring new application areas for implicit systems. That is, exploring what new actions and activities systems might engage in rather than simply automating existing ones. One major resource in this work has been design research, an area that has pioneered thinking about and approaching what systems can do in new ways. As Kelley, one of the founders of IDEO puts it, "enlightened trial and error outperforms the planning of flawless intellect (Winograd 2006)." So rather than set out with clear sense of what our systems will do, we are attempting to instead test and explore how implicit systems might work in a design led way. More broadly, our research goals can be broken down into three potential contributions:

1. Surveying and challenging existing user interactions with ubiquitous and smart technology to expose design opportunities.
2. Understanding Machine Learning as an actual limited part of systems that can be approached and shaped by designers and users.
3. Unpacking the social implications of implicit interactions across information, interfaces, and infrastructures.

While both our overarching project and this design research are at early stages, we are approaching our developing design process and artifacts themselves as ways to acquire new knowledge (Zimmerman, Forlizzi, and Evenson 2007). In this position paper, we outline and detail the progress of three such methods that correspond to each of the potential contributions, and share resulting reflections and questions that contribute to the design of meaningful and appropriate user experiences of Artificial Intelligence. In the first method, we have explored the creation of a design workbook to map varied conceptual approaches and definitions of implicit interaction. In the second method, a workshop on designing with data was employed to explore and understand how data can be used in novel ways. Lastly, in the third method we have developed a simple system that rethinks a technical notion ("leaky abstractions") to explore new types of system behavior.

A Design Workbook on Implicit Interaction

Our first approach has been the creation of a *design workbook* to collaboratively unpack definitions and implications of implicit interaction while exploring opportunities for intelligent system action. A design workbook is a collection of design concepts, proposals and related material that creates a design space in which participants can engage with or expand upon design ideas, issues, and investigations (Gaver 2011). While design workbooks can be beneficial for designers working alone or in teams, its recognition that complex designs emerge slowly and often through the synthesis of tacit relationships between an array of concepts, affords its position as a boundary object for multidisciplinary teams and in particular communicating the intellectual rigor of design (Gaver 2011; Wolf et al. 2006).

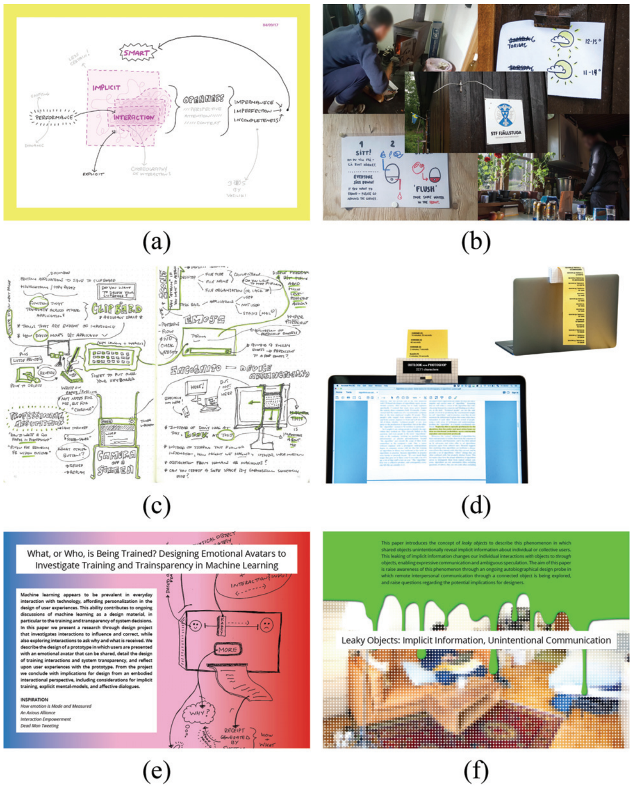
As our project work is comprised of multiple academic disciplines from differing philosophical and methodological backgrounds (i.e. Artificial Intelligence, Social Sciences, and Interaction Design), our design workbook serves as a design space in which intentions, objectives and aspirations can be communicated and aligned. Ultimately, as Interaction Design strives to unpack and overcome barriers of designing novel and consequential products and services with and for Artificial Intelligence, we are equally interested in exposing the black box of design for participation and collaboration.

Our design workbook is composed of five sections. The first section *Implicit: Meanings, Definitions, Terms* is a collection of words from meetings, workshops and emails that have been used to describe or define implicit interaction. The content of this section has been particularly important in challenging prior definitions of implicit while revealing disciplinary assumptions and mental models through subsequent card sorting exercises. The second section *Examples: Interactions, Services, Systems* is a visual collection of projects that both inspire and provoke while more importantly affording concrete examples for colleagues to reference during project activities. The third section *Domains: Situations, Contexts, Opportunities* is another visual collection, yet of problem spaces, complex challenges and interesting areas that prompt ideation and foreground an alignment in meaningful real-world applications. The fourth section *Technology: Data, Activations, Inferences* is a list of existing and aspirational data streams and sources that has been a key starting point in latter engagements with data as a design material. The fifth and final section is *Projects: Concepts, Abstracts, Briefs* and serves as a working portfolio of completed and potential projects from speculative academic abstracts to utilitarian ideas to disturbing provocations.

One example of such a provocation is the project brief written for *Designing and Prototyping a Pee-ometer to Investigate Training in Machine Learning*:

> Machine Learning is increasingly prevalent in everyday interactions with technology, affording personalization and prediction in the design of user experiences. This ability contributes to ongoing discussions of Machine Learning as a design material, in particular to the explicit and implicit training of system decisions. This project investigates interactions to initiate, influence, and correct machine learning while reflecting upon the user experience of engaging in machine training. How could and should we enable users to train and re-train Machine Learning algorithms? And how might user training of algorithms in turn intentionally or unintentionally train users?
>
> This project explores these questions through the design and prototyping of a *pee-ometer*, a connected wearable that predicts when a user has to pee based on body movements. Following foundational research, design workshops and cultural probes that investigate the training of non-technological objects, people and animals, a *pee-ometer* with a tangible user interface will be designed and prototyped to predict pee habits, suggest user actions and respond to user training.

While this project brief is obviously not advocating that there should be *pee-ometers*, by conceptually surfacing and potentially prototyping the possibility of such a device, working on the brief simultaneously reveals social tensions, relational frictions and interactional loops with smart technology while inviting those working on it to extend technical practices, such as training, into the design space. Thus, as the project navigates multi-disciplinary collaborations and investigates novel intelligent systems such as *semantic avatars* (Nilsson, Sahlgren, and Karlgren 2016), our design workbook serves as an arena for participation, critique and discourse.



(a) (b) (c) (d) (e) (f)

What, or Who, is Being Trained? Designing Emotional Avatars to Investigate Training and Trainsparency in Machine Learning

Machine learning appears to be prevalent in everyday interaction with technology, affording personalization in the design of user experiences. This ability contributes to ongoing discussions of machine learning as a design material, in particular to the training and transparency of system decisions. In this paper we present a research through design project that investigates interactions to influence and correct, while also exploring interactions to ask why and what is received. We describe the design of a prototype in which users are presented with an emotional avatar that can be shared, detail the design of training interactions and system transparency, and reflect upon user experiences with the prototype. From the project we conclude with implications for design from an embodied interactional perspective, including considerations for implicit training, explicit mental-models, and affective dialogues.

INSPIRATION
How emotion is Made and Measured
An Axious Alliance
Interaction Empowerment
Dead Man Tweeting

This paper introduces the concept of *leaky objects* to describe this phenomenon in which shared objects unintentionally reveal implicit information about individual or collective users. This leaking of implicit information changes our individual interactions *with* objects to *through* objects, enabling expressive communication and ambiguous speculation. The aim of this paper is raise awareness of this phenomenon through an ongoing autobiographical design probe in which remote interpersonal communication through a connected object is being explored, and raise questions regarding the potential implications for designers.

Leaky Objects: Implicit Information, Unintentional Communication

Figure 1: Example pages from design workbook, including a) sketch diagram of smart implicit interactions, b) photos of outdoor domain opportunities, c) annotated sketches following multidisciplinary workshop, d) concept investigating screenshots as a data source e) fictional abstract of emotional avatars, and f) abstract of in-progress project Leaky Objects

A Workshop on Designing with Data

Our second approach has been to investigate through design workshops what diverse data sources might mean and how they can be used to think about implicit system action. A growing body of research in the HCI Design Research community has been investigating data as a design material (Brown, Bødker, and Höök, 2017; Dalton et al. 2017; Boucher and Gaver 2017), i.e. a material that is approachable and shapeable by designers and possibly end users. Within our current work on implicit systems, data as a design material can be more specifically expressed as something that enables system action without that action being necessarily well defined. Indeed, from the perspective of building AI (or a Machine Learning model), data is an absolute requirement. We cannot learn anything if there is no learning material available. Data for a Machine Learning model is typically connected to a task the model is supposed to perform. If we want to categorize images, then we need labelled images to learn from. If we want to classify sentiment in text, then we need text examples of how the various sentiments are expressed. Thus, more traditional approaches to designing with data often focus on clear applications of what a system needs to do. For example, in some cases training data is collected and used to train systems which can then engage in the task unguided. We, instead, opted for what might be perceived as a backwards approach, starting with data as a material from which to ideate potential use cases, application domains, and system activities.

Our design process began with self-data collection in which screenshots from the authors' computers were taken every minute over a six-week period of time. While we wrote a program that utilized Google's image recognition API to convert these screenshots to text, we decided in parallel to inquire into the conceptual properties and arrangements of the gathered data by using a framework of *materials experiences* to investigate the practices, or situated 'ways of doing', between people and data (Giaccardi and Karana 2015). For our first workshop on designing with data, a script was used to randomly generate 'booklets' of data from the screenshot database of the first author for each of the other five workshop participants. Each booklet of data consisted of 20 screenshots from varying time intervals, i.e. across the entire six weeks, a week, a day, an hour and 20 consecutive minutes. The screenshots were then indiscriminately 'shaped' by the designer, or workshop leader and data owner, in which a series of predetermined filters, distortions, zoom lenses and effects were applied. At the beginning of the workshop, the data booklets were handed out to each participant to first familiarize with before handing out a series of five prompt cards to extract and map inferences, reflections and discussions from the data. The cards included questions regarding ownership, contexts, emotions, aspirations, and ecosystems. The workshop concluded with a speculative exercise in which participants were asked to imagine how different actors, from specific colleagues and technologies to more general personas and services, might misuse mapped inferences. Structured to design disruptions, the concluding step situated the experienced properties and performances of the data in external and consequential contexts.

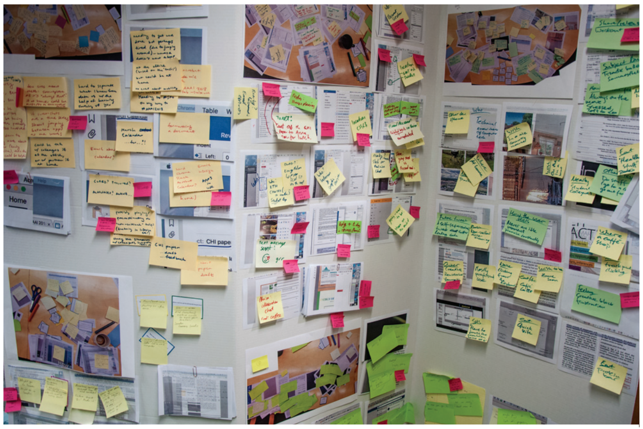


Figure 2: Screenshot 'booklets' with inferences, reflections, and discussion points from workshop participants

Prior to the workshop, our application ideas and directions for the captured data centered on actions such as advertising and recommendations that could be based on text extracted from the screenshots. Through the materialization of data and by taking an unconventional approach relative to the development of Machine Learning models, we were able to open a design space regarding how this data could be used to present more complex representations and aspects of users in new and different ways. For example, our subsequent conceptual directions that are driving current ideation included activity and inactivity hierarchies, behavioral adjustments in response to data tracking, enhancing rather than obscuring, social traces of data sharing, and the pacing of rhythms and routines. Therefore, investigating data as a material to understand the strange and perhaps even hidden aspects of online and computer based activity has enabled us to reimagine new possibilities of how systems might approach data through activities centered on how humans make sense of data.

A Prototype on Asymmetrical Interactions

Our third approach has been the prototyping of a simple informational infrastructure, or a custom Internet of Things application, to understand and design counter-strategies for asymmetrical interactions of data-driven systems in use. Prototyping is an established, interdisciplinary method employed by design researchers and interaction design practitioners for multiple purposes including but not limited to understanding an intended experience (Buchenau, Francisco, and Suri 2000). While prototypes can also reveal potential implications of proposed products, services and systems, it is less clear how designers might engage with the underlying informational infrastructures of data-driven devices and applications, such as those supported by Artificial Intelligence, to not only expose but also transform their functioning. This engagement by designers to materialize or open up an infrastructure for either design or local user intervention (Davoli and Redström 2014), is of particular interest to our work regarding conflicts of agency and concerns regarding privacy. Therefore, in addition to design explorations into new application areas, an ongoing prototype in which we are investigating the materialization of an everyday data-driven infrastructures is the autobiographical design probe Leaky Objects (Helms 2017).

Prompted by a change in communication patterns observed by the first author of this paper, the design probe initially intended to investigate how people might indirectly communicate with shared things about each other. Following the deployment of simple sensors within a domestic context and the development of a custom web application in which the status of these sensors could be requested from an Arduino, the prototype next sought to overcome obvious asymmetries in agency by incorporating a mechanism to reveal when sensor information is accessed. For example, as one sensor is a photocell attached to a floor lamp that checks the status of the light, a custom power-switch was appropriated into an awareness indicator, causing the light to flicker when its status is remotely requested. While the prototype introduced the concept of *leaky objects*, a playful reimagining of the computer science notion *leaky abstraction*, to describe the phenomenon in which shared objects leak implicit information that results in unintentional communication, it additionally surfaces the potential for further investigations into counter-strategies of obfuscation as the inherently unfinished and messy nature of a prototype creates an opening for the design of further interactions, appropriations, and hacking.

While we have used the prototyping of a simple informational infrastructure as a design method to investigate the potential social implications of implicit interactions in data-driven systems, we also hope to engage interaction designers in discussions on potential strategies of approaching the complex challenges of asymmetry in concerns of agency and privacy. As we continue to engage with more complex and layered data streams that afford Artificial Intelligence and Machine Learning techniques to support implicit interactions, we plan to continue an increased engagement in prototyping as a method for the design of meaningful and responsible user and system interactions.

Symposium

We will share our work in a 20-minute presentation format.

Author Biographies

Karey Helms is a PhD student at KTH. **Barry Brown** and **Airi Lampinen** are faculty members at Stockholm University. **Magnus Sahlgren** is a Senior Scientist at RISE SICS and at the Swedish Defense Research Agency.

Acknowledgements

This work is funded by the Swedish Foundation for Strategic Research (RIT15-0046).

References

Boucher, A. and Gaver, W. 2017. Designing and Making the Datacatchers: Batch Producing Location-Aware Mobile Devices. In *Proceedings of Tangible, Embedded, and Embodied Interaction*, 243–51.

Brown, B.; Bødker, S.; and Höök, K. 2017. Does HCI Scale? Scale Hacking and the Relevance of HCI. *Interactions*, 24(5), 28–33.

Buchenau, M.; Francisco, I.S.; and Suri, J.F. 2000. Experience Prototyping. In *Proceedings of the Conference on Designing Interactive Systemss*, 424–33.

Schnädelback, H.; Jäger, N.; Nabil, S.; Dalton, N.; Kirk, D.; and Churchill, E. 2017. People , Personal Data and the Built Environment. In *Proceedings of the Conference on Designing Interactive Systems*, 360-363.

Davoli, L. and Redström, J. 2014. "Materializing Infrastructures for Participatory Hacking. In *Proceedings of the Conference on Designing Interactive Systems*, 121–30.

Gaver, W. 2011. Making Spaces: How Design Workbooks Work. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems,* 1551–60.

Giaccardi, E. and Karana, E. 2015. Foundations of Materials Experience: An Approach for HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2447–56.

Helms, K. 2017. Leaky Objects : Implicit Information, Unintentional Communication. In *Proceedings of the Conference on Designing Interactive Systems,* 182–86.

Ju, W. and Leifer, L. 2008. The Design of Implicit Interactions: Making Interactive Systems Less Obnoxious. *Design Issues* 24 (3): 72–84.

Nilsson, D.; Sahlgren M.; and Karlgren, J. 2016. Dead Man Tweeting. In RE-WOCHAT Workshop on Collecting and Generating Resources for Chatbots and Conversational Agents: Development and Evaluation.

Winograd, T. 2006. Shifting Viewpoints: Artificial Intelligence and Human-Computer Interaction. In *Artificial Intelligence* 170 (18): 1256–58.

Wolf, T.; Rode, J.; Sussman, J.; and Kellogg, W. 2006. Dispelling 'design' as the Black Art of CHI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 521.

Zimmerman, J.; Forlizzi, J.; and Evenson, S. 2007. Research Through Design as a Method for Interaction Design Research in HCI Design Research in HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 493–502.

Designing Therapeutic Care Experiences with AI in Mind

Aisling Kelliher
Department of Computer Science
Virginia Tech
Blacksburg, VA, 24060, USA
aislingk@vt.edu

Barbara Barry
Center for Innovation
Mayo Clinic
Rochester, MN, 55902, USA
Barry.Barbara@mayo.edu

Abstract

Designing systems and services with AI functionality as part of a care experience presents a range of challenges and opportunities. Limitations with sparse or missing data can make algorithmic training difficult, while the opaqueness of some black box methods muddies the process of interpreting outcomes. Human expertise and knowledge need to be carefully integrated at appropriate stages to inform both the AI approach and the fulfillment of the overall care cycle. Tackling this complex problem space requires a multidimensional and multi-stage approach integrating technical, social, medical, design and HCI knowledge. Based on our work creating therapeutic AI systems for cognitive and physical training, we propose six key system design challenges for consideration.

Introduction

Over the next decade, artificial intelligent technologies are expected to achieve unprecedented awareness and understanding of people (Stone 2016). While the timetable and full extent of these expectations may vary (Brooks 2017), as designers, we are clearly at an important juncture in terms of grappling with AI as an increasingly significant form of design material (Holmquist 2017). In recent years, we have engaged with this material within the context of designing and deploying therapeutic systems for mental and physical wellness and healing. Our work is focused less on making machines that care or do caring tasks, and more on conceptualizing and orienting the entire care experience from the person's point of view - with AI in mind. This means considering the diversity of human actors involved in creating and experiencing AI health systems, including system designers, patients, doctors, caregivers, and family members. It also involves consideration of the perceived impact of AI systems; physically, socially, and personally.

Building on this approach and our experience working within mental health and rehabilitation contexts, we propose a number of issues that we believe are important for AI wrangling designers to consider and address. We review two cases of our work in related health care domains, highlighting incidents and issues encountered therein, and derive an initial set of questions for consideration when designing with AI in mind.

Design Cases

Interactive Neurorehabilitation for Stroke

Stroke is a leading cause of serious long-term disability in the United States and the most common neurological disorder worldwide (Benjamin 2017). While physical therapy training has demonstrated increased likelihood of recovery (Krakauer 2005), the realization of such therapy in the clinic over long periods of time is difficult for multiple reasons including availability of facilities and experts, financial cost, and the intense patient effort required (multiple times a week for several years). In response, home based, patient administered approaches have emerged as a potential viable solution, which can be effective in conjunction with therapy in the clinic or even as the primary mode of therapy (Anderson 2002).

Developing automated or semi-automated healthcare systems for unsupervised or lightly supervised use in the home presents multiple personal, technical, and design challenges (Baran 2015). Primary issues include patient adherence; recreating a supervised therapist experience without the therapist present; and system constraints, including system size, system complexity and robustness, and home privacy intrusion. While automated therapy in the home is a future end-goal for AI based systems, for now, semi-automated approaches are currently most appropriate, whereby the therapist visit occasionally in person or by video conference to evaluate patient progress and evolve the therapy protocol. In response to these challenges and realistic constraints, we are currently developing the HOMER system, which uses custom designed therapy objects, a combined computer vision and machine learning approach, and an interactive tablet interface to administer an adaptive training protocol (Kelliher 2017).

For our system to work, we need to be able to semi-automatically and accurately measure and assess patient movement quality while they are engaged in therapy activities in the home. However, developing computational agents to assist with this need is hampered by two significant factors. First, there is little readily available patient data to train a system, while second and more fundamentally, there is a lack of consensus among physical therapists regarding the standardized, quantitative evaluation of movement quality components and the influence of such components on overall functional ability (Levin 2009). In practice, therapists typically select which components to focus on based on their individual and collective experience and training, rather than a standardized ontology of component level labels for movement quality (Wolf 2001). These two factors combine to make it very challenging for a technological rehabilitation system (whether supervised or unsupervised) to reproduce both a complex therapy experience and a reliable approach for movement quality assessment.

From a design perspective, it is also vital that our system be accepted by the patient and/or the caregiver, meaning the system needs to occupy a small physical footprint, be straightforward to use and maintain, provide accurate and helpful feedback, and above all, to assist in motivating the patient to adhere to the training schedule and protocol. Our light weight tabletop system consists of a custom fit mat, 6 customized therapy artifacts and their container, a table mounted depth camera and mini-computer module, and a tablet device with a custom web application (see Fig 1.). This system can easily fit temporarily or semi-permanently on a kitchen table or spare room desk, and is designed for straightforward assembly, power charging, and data download. The feedback approach can be adapted to the abilities and progress of the patient (e.g. more lenient for moderately impaired or when the patient is fatigued).

The form and function of the objects in our system requires design consideration of the inter-relationships between the perceived affordances of the objects, the goals of the therapy protocol, the ability of the computational components of the system to capture the participant activity, and the desired therapy outcome with respect to everyday life activities. As such, the set of objects in our system (see Fig. 1b) are designed to support cross-mapping, problem solving, and generalizable activity strategies through their open-ended affordances, combinatorial possibilities, and perceived correlation with diverse artifacts of daily living (e.g. pushing a button, using an iron, writing with a pen, turning a key etc.)

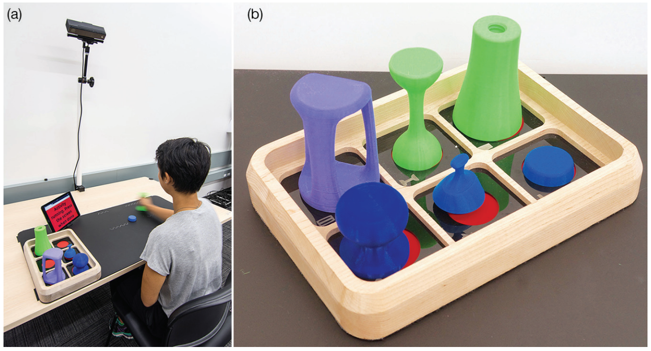


Figure 1. a) The interactive stroke rehabilitation system including mat, objects, tablet and mounted camera; b) set of 6 3D printed therapy objects

Creating functional and compelling interactive home based therapeutic systems requires a participatory and iterative design approach. Introducing sensing and control technologies (e.g. cameras and wearable sensors) into the home necessitates direct conversations between designers and home dwellers as to the nature of the data captured, access to that information, and transparency about how the AI components of the system are trained to potentially interpret it. In addition, the strength of the system is in the potential for knowledge and growth in both the human and computational agents as the system is tried out, refined, and improved based on the quality and subsequent analysis of the quantitative and qualitative data collected.

Digital Mental Health Futures

Functional brain imaging has been useful in mapping the neural circuitry of psychiatric disorders and promises a new understanding of the underlying neural mechanisms of psychotherapy with implications for identifying the most effective treatment for an individual (Linden 2006). Drawing on this research and an analogy to optogenetics, the controlled use of light to activate specific neurons, we speculated about creating an AI that could tailor talk therapy sessions by learning the most effective therapeutic techniques for an individual's experiential and neural response (Barry 2009). In our wildest imaginations, we envisioned that an open source collection of therapeutic techniques could also help the psychiatric community track biological evidence and patient preferences for or against any given therapeutic technique.

We built an initial prototype and ran an exploratory study to examine the idea of using machine learning to create the most efficacious therapy session for an individual. The AI "therapist" followed a standardized therapeutic protocol. First, it surveyed study participant communication preferences and anxiety levels. Then, it assembled and delivered a tailored therapy session as sequential units of therapeutic techniques delivered via audio. The therapeutic units guided the participant to reflect on anxiety reinforcing behaviors and learn new techniques for anxiety reduction. The AI measured participant anxiety levels after each unit of therapy and then optimized the session for content that reduced anxiety. We did not incorporate brain imaging into this speculative design exploration. We did engage in discussions with mental health professionals, developers, designers, and study participants about the possible implications of feedback loops between patients, AI, fMRI, and a therapist working in concert to treat psychiatric disorders.

During debriefing discussions with 32 study participants, 29 considered the AI helpful overall and completed their session with lower levels of anxiety than when they began. The three participants with rising anxiety cited cognitive overload of therapeutic techniques or were annoyed by the voice of the AI therapist. Some participants were intrigued by the idea of an AI therapist being more "neutral" than a human one and by a real-time feedback system that responded to their emotions. Others identified possible divergence between what a patient, the AI, and a therapist might consider the "best" set of therapeutic techniques. Mental health professionals questioned the algorithm responding to anxiety interval measures because an immediate rise in anxiety may mean a therapeutic technique is uncomfortable but not necessarily ineffective. Ethical issues about trusting AI system intentions and concerns about AI monitoring of mental health and brain activity were expressed.

Design issues emerged through use of our speculative prototype that call out tensions between biological health, the lived experience, and what it means to be understood by a therapist, whether AI or human. We advocate that speculative designs be used to generate possibilities and identify risks for AIs as participants in therapeutic treatments, especially to help ensure that AIs are well designed to meet the needs of patients before they are introduced into care experiences.

Design Questions

In reflecting on our design cases we identified six key questions for designers to consider as AIs grow in their complexity and capability. In exploring these questions, as a design community, we can observe how AIs understand and respect the person's point of view.

> *How does human behavior, captured and analyzed and interpreted by AIs influence care opportunities and decisions?*
>
> *How, or should, humans and AIs reach consensus on interpretations of data (when sometimes even humans can't agree)?*
>
> *How are both personalization and scalability redefined and designed in an era of big data, missing data, and sparse data?*
>
> *How should we design autonomous and semi-autonomous systems that provide therapeutic value and will be anticipated, accepted, and embraced by human actors in diverse environments?*
>
> *How should AIs be designed, adapted, and regulated as trusted members of care teams?*
>
> *How can design help identify, anticipate, and address ethical issues that may emerge when AIs are involved in care?*

We believe that mindful consideration of these questions teams is particularly important in healthcare contexts where complex issues concerning emotions, power, inclusion, decision making, and responsibility are key human variables. Working with the powerful material of AI in such environments presents the potential for tremendous advancement as practiced within a reflective and careful design framework.

Author Bios

Aisling Kelliher is an associate professor of Computer Science at Virginia Tech, with joint appointments in the School of Visual Arts and the Institute for Creativity, Arts, and Technology. Aisling co-leads the Interactive Neurorehabilitation Lab at VT, where she works with an interdisciplinary team of designers, physiotherapists, computer scientists and engineers developing light-weight, cost-effective systems for conducting semi-supervised stroke rehabilitation in the home. She is also a Co-PI in the newly formed Synergistic Musculoskeletal Adaptive Research and Technology Lab (SMART Lab), a joint initiative between the Virginia Tech Carillion School of Medicine and Carilion Clinics. The SMART Lab will investigate the impact that pain, disability, and pathology have on individuals across the lifespan through the design and development of prevention and post-injury intervention programs and systems.

Barbara Barry is the Design Strategist for the Mayo Clinic Center for Innovation and an Assistant Professor in the Mayo Clinic School of Medicine. She is an interdisciplinary research scientist who uses applied anthropology and computer science to fuel innovation in industry, public and humanitarian sectors. She has led in-depth human-centered design projects for Mayo Clinic to improve the health of young adults and co-designed patient-centered care models for emerging markets. Prior to joining Mayo Clinic, she worked with neuroscientists and psychiatrists to develop personalized digital mental health apps and led UN funded programs to understand how technology can scale education and health care interventions to help children displaced by conflict and natural disasters. Barry has a Ph.D. and M.S. from Massachusetts Institute of Technology and a B.F.A. from Massachusetts College of Art and Design.

References

Anderson, C., Ni Mhurchu, C., Brown, P., and Carter, K. 2002. Stroke rehabilitation services to accelerate hospital discharge and provide home-based care. *Pharmacoeconomics,* 20(8): 537–552, 2002.

Baran, M., Lehrer, N., Duff, M, Venkataraman, V., Turage, P., Ingalls, T., Rymer, Z., Wolf, S., and Rikakis, T. 2015. Interdisciplinary concepts for design and implementation of mixed reality interactive neurorehabilitation systems for stroke. *Physical therapy,* 2015;95:449-460

Barry, B. 2009. Metatherapy: Designing Open Source Software for Mental Health Research and Care. White paper. MIT Media Lab

Benjamin E.et. al. 2017. Heart Disease and Stroke Statistics – 2017 Update: A report from the American Heart Association, *Circulation,* Volume 135, Issue 10, pp 146 – 603

Brooks, R. 2017. The Seven Deadly Sins of AI Predictions, *MIT Technology Review*, October 6, 2017.

Holmquist, L. 2017. Intelligence on tap: artificial intelligence as a new design material. interactions 24, 4 (June 2017), 28-33.

Kelliher, A., Choi, J., Huang, JB, Kitani, K., and Rikakis, T. 2017. HOMER: An Interactive System for Home Based Stroke Rehabilitation. In *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '17*). ACM, New York, NY, USA, 379-380.

Krakauer, J. 2005. Arm function after stroke: from physiology to recovery. In *Seminars in neurology*, volume 25, Number 4, pp 384–395.

Koch, J. 2017. Design implications for Designing with a Collaborative AI. The AAAI 2017 Spring Symposium on Designing the User Experience of Machine Learning Systems Technical Report SS-17-04

Levin, M., Kleim, J, and Wolf, S. 2009. What do motor recovery and compensation mean in patients following stroke? *Neurorehabilitation and Neural Repair*, 23(4), pp 313 – 319.

Linden, DEJ. 2006. How psychotherapy changes the brain – the contribution of functional neuroimaging. *Molecular Psychiatry*, 11, 528–538

Stone, P et al. 2016. "Artificial Intelligence and Life in 2030." One Hundred Year Study on Artificial Intelligence: Report of the 2015-2016 Study Panel, Stanford University, Stanford, CA, September 2016. Doc: http://ai100.stanford.edu/2016-report. Accessed: September 6, 2016.

Wolf, S., et al. 2001. Assessing wolf motor function test as outcome measure of research in patients after stroke. *Stroke,* 32(7): 1635-1639.

Insectile Indices: Los Angeles, 2027

Yeawon Kim
Media Design Practices, Art Center College of Design, Pasadena, CA
ywonkim89@gmail.com

Abstract

Crime prediction technology – we have seen it in the movies, but what has in the past been pure fiction is now quickly becoming a reality. *Predpol, HunchLab* and *ComStat* are all different types of relatively new crime prediction software, or "predictive policing" software, that demonstrate how algorithms and other technologies can be used within urban infrastructures to predict crime. However, utilizing these technologies and algorithms to collect data to predict crime, which is invariably subject to and tainted by human perception and use, can lead to a number of adverse ethical consequences – such as the amplification of existing biases against certain types of individuals based on race, gender or otherwise. On the other hand, if data can be gathered by some artificial intelligence (AI) means – thereby removing the human component from such data collection, can doing so result in more efficient and accurate crime prediction? Furthermore, will we in doing so also reshape the aesthetic of urban nature, especially when one takes into account the constant evolution of AI?

Introduction

Insectile Indices is a speculative design project that considers how electronically augmented insects could be trained to act as sophisticated data sensors, working in groups, as part of a neighborhood crime predictive policing initiative in the city of Los Angeles, 2027. This project is not only an investigation into the ethics of this controversial idea, but also an aesthetic exploration into the deliberate alteration to a natural wildlife ecosystem of insects.

In 2007, the Defense Advanced Research Projects Agency (DARPA) asked American scientists to submit proposals to develop technology to create insect-cyborgs, the results of which led to a plethora of troubling and worrisome commentary. Rather than build off of a frightening narrative that discusses the potential sinister militaristic use of such technology, this project does the opposite and imagines instead an aesthetically pleasing utopia where these insect-cyborgs have social utility and work towards the public good of humanity. *Insectile Indices* also plays with the idea of aesthetics in our future techno-driven world by addressing whether we are more apt to silently "turn the other cheek" to more pervasive surveillance if these insect-cyborgs become more aesthetically pleasing to the eye.

Speculative Scenario: Los Angeles, 2027

Nowadays, we commonly encounter beautiful cyborg moths in the city of Los Angeles that secretly conduct surveillance on our daily life. Initially, research in this area was limited to robots created to imitate insect behavior, but technology was further developed to manipulate the bodies of insects for surveillance in the city. There are numerous reasons to use insects to monitor city – namely, because of an insect's high sensitivity to smell, ease in which their DNA can be modified and programmed, and their power and ability to swarm in great numbers. Designers and scientists that in this field believe that the insect contains a gigantic breadth of evolutionary experience of solving problems in both artificial and natural ecosystems and, as such is the most guaranteed source of innovation for surveillance as a result of nature's billion years of evolution. Also, it is more efficient and sustainable to grow insect bodies for mass production, rather than mass-producing robots, which require expensive materials.

Background

As mentioned above, in 2007, DARPA asked American scientists to submit proposals to develop insect-cyborgs, with the objective of somehow developing "micro air vehicles" – ultra-small flying robots capable of performing surveillance in dangerous territories. To that end, Cornell University researchers were successful in implanting electronic circuit probes into tobacco hornworms as early as in the pupae growth stage. Specifically, the hornworms passed through the chrysalis stage to mature into moths whose muscles can be controlled with the implanted electronics (Bozkurt, Gilmour and Stern 2008). Various insect species such as dragonflies, beetles, cockroaches and crickets were used for Cornell's research in creating the cybernetic organism. The lifespan of the resulting "cybug" also increased through converting heat and mechanical energy that the insects naturally generated (Aktakka et al. 2008).

Twenty years after this research, this technology has trickled down to everyday life in the city of Los Angeles. Cybug farms are prevalent in the suburbs of Los Angeles, and produce up to 70 million Cybugs a year. AI manages the quality control of these Cybug breeds – for example, ensuring caterpillars suitable for surveillance and supporting the successful metamorphosis of these insects to create healthy Cybugs. Full-grown moth Cybugs are freshly stored at special designed vehicles in low temperatures, and are safely transported to the Cybug Hotel in Los Angeles.

Even though the AI system and its predictive policing algorithms intentionally omits certain data points in order to eliminate existing ethical biases, AI alone still cannot fully avoid learning the bias from data suggested from humans. The Cybug is able to avoid this. For example, the Cybug substitute initial police inspections by using its biological sensors to gather crime data and sends it to the Cybug Hotel. And, it makes core decisions on behalf of the police, such as where to deploy more police officers, Cybugs, or how to control navigation to crime scenes for efficiency. Therefore, Cybugs and the police department accomplish their objectives through "symbiotic autonomy" to substitute the unnecessary labor of police inspection with Cybugs.

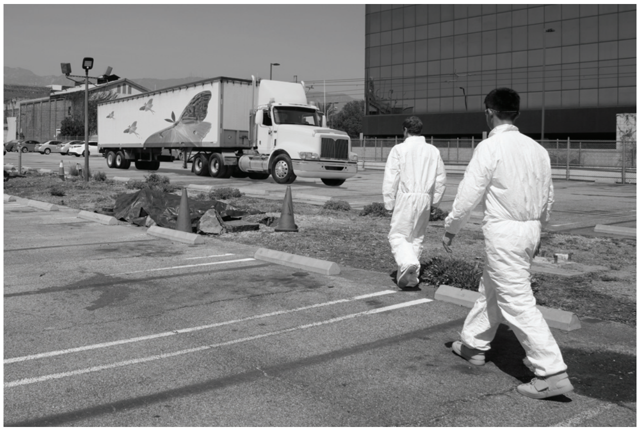


Figure 1: The Cybug Policing Vehicle, 2027

Cybug Hotel

The Cybug Hotel is the AI center that manages the city infrastructure. It analyzes data foraged by the Cybugs and controls their movements. All the data gathered are stored as memory by the Cybug Hotel. And, the Hotel sends feedback to operate urban infrastructures such as streetlights, mass transportation, and the police. As a result, every system in the city is interconnected because of the actions of the insect cyborg.

The Cybug Hotel system analyzes the data, such as the pattern of events in the city, and determines the optimal route and method for the Cybugs to investigate neighborhoods. This interaction between different moth Cybugs and the Cybug Hotel is based on electric signals that help them react rapidly on the constant change, stream and influx of data. The Cybugs rotate back to the Hotel every four to six hours to recharge its energy and to report data to the AI system.

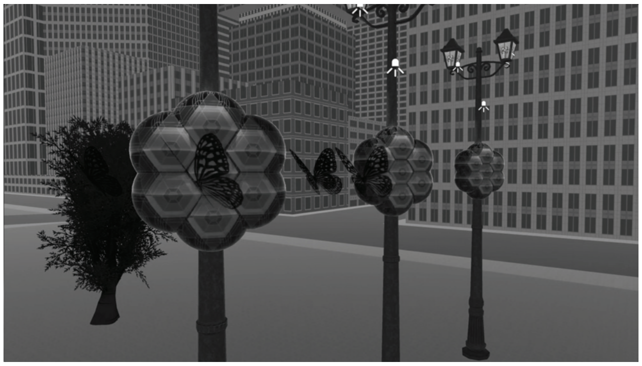


Figure 2: The Cybug Hotel, 2027

Jobs

The Cybug infrastructure has also created numerous jobs, such as the Cybug gardener, analyst and collector. The gardeners are specialized to breed the insect body and inspect the biological process of the growth, so that every insect satisfies Cybug standards for deployment to the city. Cybug analysts keep track of the evolution of the species, and limit the interaction between wild and genetically modified Cybugs. And, more importantly, they analyze the data aggregated by the Cybug Hotel, which was initially collected by the Cybugs. Finally, the Cybug collectors are responsible for gathering dead Cybugs so that it keeps track of their evolutionary processes, and so their remains can be recycled.

The Species

The form and pattern of the Cybug body is developed by AI system. The method used to design the pattern of the moth Cybug wings is based on "Image Hallucination" - image recognition produced by artificial neural networks. The visual patterns generated by neural network are applied to the redesign of the insect's biological body, which is powered by algorithms that are modeled after the evolutionary process of the insect. There are three different electronically augmented moth species that are designed by the system and located at the Cybug Hotel: the Hyalophora

Cecropia moth, the Antheraea Polyphemus moth and the Lunar moth.

The Hyalophora Cecropia moth, which is a blue colored moth, gathers audio from the Los Angeles urban landscape, secretly listening or recording your voice or mechanical sounds implying problems occurring in Los Angeles. The species has been used for foraging sound data, such as conversations in intimate urban spaces such as elevators, alleys or homes to detect suspicious dialogue. While this moth used to be found as far west as the Rocky Mountains and as far north as many Canadian provinces on maple trees, it is now only produced on Cybug farms for human surveillance purposes and difficult to find in the wild due to the prevalence of these Cybug farms.

Antheraea Polyphemus, the red moth species, is used to catch images and track movements of everyday life – much like that of a CCTV camera. But, the Cybug is more effective in this regard as it camouflages into the city landscape as compared to a CCTV. This type of moth was widespread in continental North America, with local populations found throughout subarctic Canada and the United States. But, like the Hyalophora Cecropia moth, it is now largely produced in Cybug farms.

And finally, the Lunar moth detects suspicious odors. This moth is commonly used for investigating chemical compounds such as explosives, drugs and weapons, rather than its historical and evolutionary use to detect pheromones and other attractants in flowers. Lunar moths are also known for their ability to effectively swarm when needed and, as such, can effectively perceive suspicious chemical odors to help people quickly notice danger and escape.

Most mature moths in wildlife can live around one to two weeks, but the genetically modified Cybugs can live up to one month with proper electrical energies controlling the body. When the Cybug eventually dies, microprobes that were initially inserted in the body are then recycled upon death for future use at the Cybug farm.

Figure 3: Crime Scene, 2027

Swarm Behavior

The Cybugs swarm based on received data to inform of dangers and prevent crimes. There are three commonly known Cybug group swarm behaviors - trap building, flocking and synchronization, all of which are learned from the evolutionary group patterns of other wild insects and animals.

Trap building behavior is derived from the Amazonian ant species Allomerus Decemarticulatus. The trap resembles a honeycomb, but works like a web. After building the honeycomb-like structure, the Cybug secretly waits for a suspicious individual, and then traps them by swarming, which substitute the traditional police search. Flocking behavior is emulated from bird migration patterns, which improves the Cybug's efficiency of flying from one spot to another. Finally, synchronization behavior, which is derived from the fireflies' bioluminescence during mating season, allows the Cybugs to be released into the urban infrastructure with the necessary synchronized data from the Cybug Hotel to complete its mission and achieve its objectives. Ironically, all of the swarm behavior considered beautiful, making spectators to catch the moments through video and photo for social media entertainment.

Figure 4: Swarm Behavior, 2027

Conclusion

The role of the Cybug Hotel, in conjunction with Cybugs, is to efficiently make decisions for policing the city without human bias, which are determined by social structure, religious beliefs and political environments and cannot avoid subjective standards of what is right or wrong. However, this speculative scenario implies AI as part of "nature," which evolves with wild and artificial factors. Even though, the idea of "nature" and "ethic" inherently conflict with one another, the Cybug, as part of a "natural" ecosystem, attempts to reconciles this conflict by making ethical decisions instead of human.

Cybug Hotel metaphorically represents a “hotel”, which is a sterile and universalized space meant for tourism in modern society. The AI system remains as mystical place that symbolizes the identity of a neighborhood, exhibited as a government’s political product to serve the public. And, the idea of tourism doesn’t come from the ontological question “what this insect is,” but instead the functional question “what does the Cybug do for public”? AI manipulates the instinct of wild insects for crime prevention and for the public utility, rather than for its historical and natural function of pollinating flowers or evolutionary proliferation. This denies insects of their role within wild nature for another form of nature itself – namely, as artificial nature ecosystem for the sake of the society. Cybug Hotel analyzes the data from residence’s facial expression, gesture, smells and voices, which help construct a ‘seamless transition’ between AI and humans, which has the effect of making technology invisible in the urban nature. And, it is this idea of ‘seamlessness’ that connects with the fusion of man, AI and the Cybug.

References

Haraway, D. 1984. A Cyborg Manifesto: Science, Technology, and Socialist-Feminism in the Late Twentieth Century. In Simians, Cyborg and Women: The reinvention of Nature. New York: Routledge, 1991: 149-181.

Bozkurt, A., Gilmour, R., Stern, D., Lal. A. 2008. MEMS based bioelectronic neuromuscular interfaces for insect cyborg flight control. In *Proceedings of the IEEE International Conference on Micro Electro Mechanical Systems (MEMS),* 160-164.

Kac, E. 1988. Transgenic Art. *Leonardo Electronic Almanac, Vol.6*: 1071-4391.

Helmreich, S., Roosth, S., Freidner, M., 2016. Sounding the limits of life: Essay in the anthropology of biology and beyond. Princeton, New Jersey: Princeton University Press, 2016.

McLuhan, M. 1964. The Medium is the Message. Understanding Media: The Extensions of Man. New York: Signet.

The Importance of UX for Machine Teaching

Martin Lindvall
Linköping University
Sectra AB
martin.lindvall@sectra.com

Jesper Molin
Sectra AB
jesper.molin@sectra.com

Jonas Löwgren
Linköping University
jonas.lowgren@liu.se

Abstract

In this position paper, we argue that UX designers should take an increasing responsibility for the process and tools used in the generation of training data for machine learning algorithms. We provide a number of annotated examples from our UX practice within the medical imaging domain to highlight different ways that a UX approach can help to select training data set, facilitate initial generation and ensure that the final systems become self-sufficient on training data, so that the systems can efficiently improve performance over time.

Introduction

One of the most important developments from a UX perspective in the machine learning (ML) domain is that algorithms today are able to improve their performance by adding more training data. This entails that processes and tools for the generation of training data can have a large impact on the success of ML projects. In many domains, the designers of the teaching systems do not themselves hold the expertise required to create training data, which means that human-centered design methods can play a key role in building systems that aid generation of training data.

This *teaching* aspect of building machine learning systems has recently received some attention. In Simard et al. (2017) the authors emphasize the role of the teacher and their interaction with data as a key factor for building machine learning systems at scale and argue for making *machine teaching* a discipline in its own. Cramer and Thom (2017), identifying and reflecting upon the impact of design decisions on ML outcomes, pose a series of questions relating to how the role of curators and annotators affect the ultimate end-user experience.

To emphasize the role of UX practice for generating training data we will highlight some key ideas illustrated by examples drawn from our work within a specific domain: medical imaging. We will describe four interactive systems that have been created and used within digital pathology, i.e. diagnosing and reviewing digital gigapixel-sized microscopic images of tissue samples such as biopsies and surgical specimens.

These examples together describe a typical two-step process we have used when designing new ML-based systems. First, we need to bootstrap a large enough dataset so that the algorithm used in the first version of the system performs sufficiently. Second, we need to ensure that the system can collect training data automatically when it is deployed, i.e, by receiving user corrections. This will make the system self-sufficient on training data, enabling a continuous improvement of the ML model. Our four annotated examples of this process are based on our own experience as UX-designers active in the medical imaging field. Two of the examples are prototypes and two are finished products that we have either designed ourselves or followed closely.

Efficient bootstrap teaching

An early step in the creation of an ML-based system, when no prior training data exist, is to somehow create an initial dataset. For pathology images this typically consist of drawing outlines over tissue regions and classifying these. Because it is a highly specialized domain this usually means engaging pathologists, who tend to be rather expensive teachers. Since it is important to make efficient use of these individuals and their knowledge, it seems sensible to align the design of the teaching environment with their experience.

Rapid interactive segmentation A well-known semi-automatic approach to assigning categories to visual regions is an *interactive segmentation tool*. The user of such tools typically use a paintbrush-style interaction to assign areas to given categories (called ”seeds”), and while doing so, areas similar to the one marked are also assigned to the same category (McGuinness and O’Connor 2010).

When we applied a human-centered design perspective to the construction of such a tool we gained valuable insights; for our initial prototype (see figure 1), the interaction was experienced as a *trading of control between human and machine*, where the human waits for the machine response after drawing an area. After a noticeable delay, the results are received and the human can make a correction, wait again, and then repeat the process. Typically, the user would be both intrigued and annoyed by the automatic assignment of the areas that were not specifically drawn over, sometimes resulting in long back-and-forth correction cycles without noticeable progress.

In a revised version, we aimed for *rapid fine-grained interaction* where spreading would be constructed as an in-

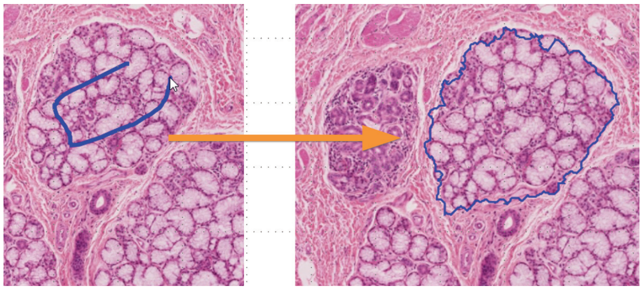


Figure 1: The initial version of our interactive segmentation tool. The user draws a path and waits for the response.

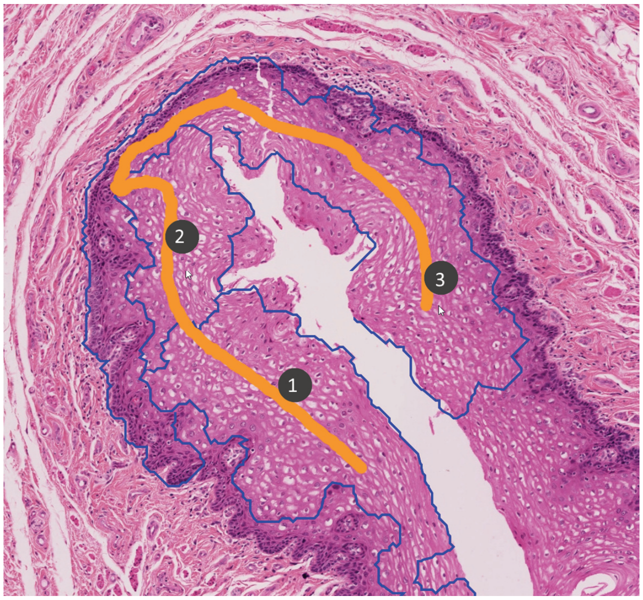


Figure 2: The revised segmentation tool. The user draws and results update in realtime, here shown for three points in time.

cremental and collaborative effort between user and system, rather than being computed slowly but accurately in every coarse-grained step (see figure 2). The tool was changed so that the threshold required for spreading increased with the distance from the original area. Additionally, we added precomputations so that results of user input typically arrive in less than 40ms, a time during which the user is not blocked from giving more input. We postulate that the more fine-grained interaction lets the user gain an intuitive understanding of the underlying mechanism and its limitations by observing many predictions over time. Overall, we believe this real-time version of the tool to be novel and much preferable to using traded control, an effect we hope to validate in future work.

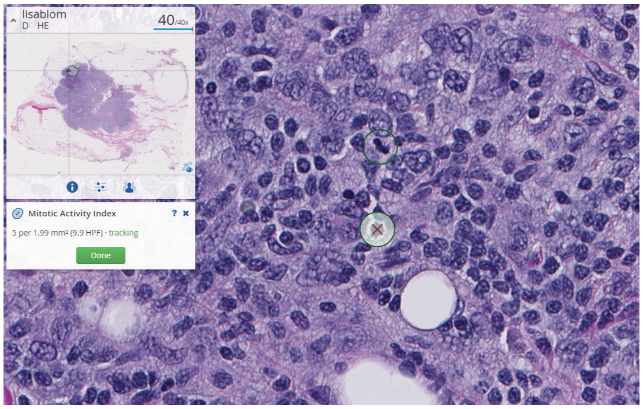


Figure 3: A manual tool to help pathologists to keep track of mitotic figures. This is used to generate training data for a future algorithm.

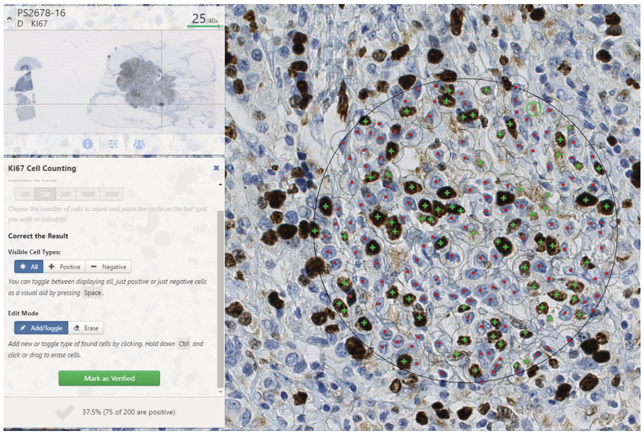
Creating intrinsic rewards Another approach to bootstrapping the initial training data set is to design a useful manual tool that generate training data as a side-effect. This approach is somewhat similar to the ESP game (von Ahn and Dabbish 2004), a two-player guessing game that created labeled training data as a side-effect of play. In the medical domain with professional users it would be inappropriate to deploy games to generate training data. Instead, the manual tool should aid the clinician in their decision making as a result of providing the tool with labels.

We have created one such tool to support pathologists in manual mitotic counting (figure 3). In this diagnostic task, the pathologist should go through ten fields of view in the highest magnification and count the number of mitotic figures. When performing this task it can be challenging to keep track of the number of mitotic figures as well as the number of fields of view. In the tool, this task is supported by keeping track of the reviewed area when navigating in the image. The user can also click on detected mitotic figures, which are then stored. Upon completion, the mitotic density can be derived using the number of stored mitotic figures and the total tracked area. Even though the tool works by rather simple means, it still turns out to be very useful for the pathologist. The side-effect is that every time a mitotic figure is clicked on, a training data example is generated. Additionally, the tracked areas that were reviewed but not clicked on can be used as examples of non-mitotic figures. By deploying this tool into a delivered product, it will generate a bootstrapping dataset of mitotic figures that can be used to train a ML-based detection system.

Designing for user corrections

Once ML systems are deployed, user corrections of the ML predictions can be used to generate additional training data. However, the UX designer needs to design specifically for this possibility. Our experience so far indicates that the most important factors for this type of design are to make sure that machine errors become apparent and that the class labels are chosen in such a way that they are easy to interact with.

This can be exemplified by ML systems used to quantify immunostains. Immunostaining is a technique used to chemically visualize protein expression in cells. A common protein used to quantify proliferation in tumor cells is KI-67.



Choose the number of cells to count and place the circle on the hot spot you wish to calculate.

You can toggle between displaying all, just positive or just negative cells as a visual aid by pressing Space.

Add new or toggle type of found cells by clicking. Hold down Ctrl and click or drag to erase cells.

37.5% (75 of 200 are positive)

Figure 4: An example of a symmetric input-output ML-system of cell counting system for Ki-67 stainings.

When using the KI-67 immunostain, the nucleus becomes brown if the cell is positive for this protein and appears blue from the background staining if it is not.

When designing an ML pipeline, two apparent choices of class labels for this problem exists: pixel labels and nuclei labels placed on the center of the nuclei. If pixel labels are used, pixels belonging to positive and negative nuclei can be visualized to the user as an overlay on top of the original image, occluding the nuclei. The user can then accept the result as is, or revert to manually counting the cells. If nuclei labels are used, the result can be visualized by placing glyphs on the center of each detected nucleus. This makes it is easier for the user to detect errors, since less ink is used to visualize the result and the original image becomes more visible. It also becomes easier to perform correction of misplaced markers since less precision is needed to click on markers than on pixels.The second approach was implemented as a product, and is shown in Figure 4.

This product illustrates the seminal principle of direct manipulation (Shneiderman 1982) that the result is presented in an *input-output symmetric way* where the user can directly manipulate the labeled data. By designing the system to allow for such direct manipulation and providing an *intrinsic reward* in terms of the actual nuclei count, user corrections can directly be used to retrain and improve the underlying machine learning model.

Another example of an ML direct manipulation interface is our patch gallery prototype shown in Figure 5, where the goal is to estimate the distribution of classes in an area. In this prototype, we generate a grid pattern over a user selected area and extract a small image patch for each point in the grid. We then feed each patch to an ML algorithm that classifies the patches into different categories, which is then shown in a sorted gallery. Each defined class in the trained model is shown as patches in the same gallery, and the user can then 1) click on a patch to see it in the main view to get a sense of its context in the tissue, and 2) change a label by either dragging the patch to the correct category or by clicking on the button or the corresponding shortcut key.

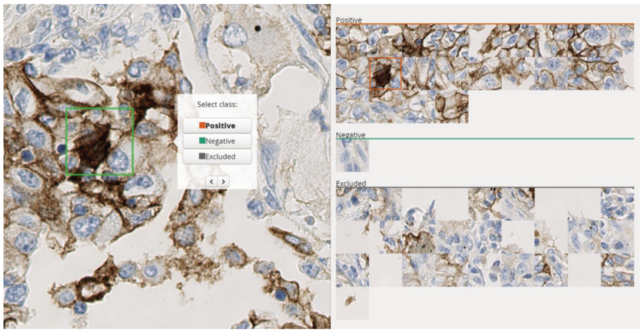


Figure 5: Patch gallery prototype, samples from the tissue is generated and classified by an ML algorithm into three classes.

Both these systems share the property with the mitotic counter in the previous section that the generated parameter can be derived from manual input only. If the nuclei detection algorithm failed to detect any nuclei, the user could still manually click on all the nuclei to calculate the KI-67 index. However, the amount of clicking would likely overwhelm the user. These user correction systems do not strictly need an ML component, but practical usability requires automated support with a certain level of prediction accuracy.

Another crucial factor when designing this type of user correction system is that the user correction accuracy needs to be higher than that of the ML component alone, in order for the generated training data to add value when retraining the ML model.

Discussion

In the design of these tools, we have paid special attention to ensuring that manual, unassisted, work-flows are preserved and as outlined in the previous section, compatible with the assisting tools. Furthermore, as the performance of models improve using the self-generating training data, we expect that our initial user interfaces need to be redesigned or augmented with interactions that are adapted to ML components with much higher performance. It is our ambition to design these so that the user can step through these ”levels of intelligence”, providing corrections and simultaneously teaching and verifying results at different levels, forming a verification staircase (Molin et al. 2016) as opposed to a steep cliff where the user has to validate all or nothing. We believe one possible way to achieve this could be to create our abstractions so that the user can always decompose a higher abstraction in terms of a lower one, an idea similar to the hierarchies of ecological interface design (Vicente and Rasmussen 1992) that we hope to explore in future work.

As pointed out by Dove et al. (2017), the interaction design community is still new to using ML as a design material. We are thus cautious to move beyond annotated examples towards more compact formulations of generative knowledge such as design patterns or principles at this stage. The examples described here may form the basis of transferable design knowledge when generalized to domain-independent visual reasoning tasks including an ML component. However, we need further studies in order to refine and validate this approach.

Conclusion

In this paper, we presented a number of annotated examples of how to manage training data generation from a UX perspective. The pattern emerging from these examples is that many of our ML projects become two-step processes. First, a training data set is created so that the initial trained ML model can reach an accuracy that will be acceptable to early-adopter users. Then by using different data collection methods designed into the first version of the product, it becomes self-sufficient in terms of training data. This allows the product to improve over time. As this process continues, the ML component will at some point become so good that the initial user interface might no longer be valid, and needs to be adapted to an ML component that performs on a higher level. How this is done is a promising area of further research. Our current plans involve a systematic explorative design effort of automation performance in the design of human-automation collaboration for visual reasoning tasks. Hopefully this will lead us toward the abstraction of genre-related generative design knowledge.

Looking at ML-based product development from the view of training data generation, we can learn that decisions made by the UX designer have an enormous impact on project success. Each step of training data generation needs to get the motivations right so that users are willing and able to provide corrections. The choice of what the training data set should consist of and thus what the ML model should predict is tightly connected to how the user interface should look, behave and be interacted with.

We challenge all UX professionals to take charge of the ML development cycle to make use of this powerful technology in the medical domain.

Biography

Martin Lindvall. Martin is an industrial Ph.D student at Linköping University exploring interaction design using machine learning as a material for creating effective ensembles of skilled medical practitioners and AI. Martin's background includes a M.Sc in Cognitive Science and ten years of experience designing and developing medical information systems as senior research engineer at Sectra.

Jesper Molin is research scientist and UX designer at Sectra exploring and designing ML-based tools used within clinical routine pathology. Jesper's background includes a M.Sc in *Applied physics and electrical engineering* and a now almost finished Ph.D in Human-Computer Interaction from Chalmers University of Technology.

Jonas Löwgren is professor of interaction and information design at Linköping University, Sweden. His expertise includes collaborative media, interactive visualization and the design theory of the digital materials.

Acknowledgments

This work was partially supported by the Wallenberg Autonomous Systems and Software Program (WASP).

References

Cramer, H., and Thom, J. 2017. Not-So-Autonomous , Very Human Decisions in Machine Learning : Questions when Designing for ML. Technical Report SS-17-04, The AAAI 2017 Spring Symposium on Designing the User Experience of Machine Learning Systems, Stanford Univ.

Dove, G.; Halskov, K.; Forlizzi, J.; and Zimmerman, J. 2017. UX Design Innovation: Challenges for Working with Machine Learning as a Design Material. In *CHI '17: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 278–288. ACM.

McGuinness, K., and O'Connor, N. E. 2010. A comparative evaluation of interactive segmentation algorithms. *Pattern Recognition* 43(2):434–444.

Molin, J.; Woźniak, P. W.; Lundström, C.; Treanor, D.; and Fjeld, M. 2016. Understanding design for automated image analysis in digital pathology. In *Proceedings of the 9th Nordic Conference on Human-Computer Interaction*, 58. ACM.

Shneiderman, B. 1982. The future of interactive systems and the emergence of direct manipulation. *Behaviour & Information Technology* 1(3):237–256.

Simard, P.; Amershi, S.; Chickering, M.; Edelman Pelton, A.; Ghorashi, S.; Meek, C.; Ramos, G.; Suh, J.; Verwey, J.; Wang, M.; and Wernsing, J. 2017. Machine Teaching: A New Paradigm for Building Machine Learning Systems. Technical Report MSR-TR-2017-26, Microsoft Research.

Vicente, K. J., and Rasmussen, J. 1992. Ecological interface design: theoretical foundations. *IEEE Transactions on Systems, Man, and Cybernetics* 22(4):589–606.

von Ahn, L., and Dabbish, L. 2004. Labeling images with a computer game. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, 319–326. New York, NY, USA: ACM.

Trees of Knowledge: Designing with Artificial Intelligence in the Urban Landscape

Xiaoxuan Liu and Godiva Veliganilao Reisenbichler
MFA Candidates in Graduate Media Design Practices at ArtCenter College of Design
sallyliu.sam@gmail.com and godivareisen@gmail.com

Abstract

In our ongoing speculative design project entitled *Topos*, we propose a public-facing, tangible user interface (TUI) that makes legible and accessible the AI systems embedded in near-future urban landscapes. By imagining AI as a public service, *Topos* interrogates the creation of public trust between people and AI systems through the medium of physical structures in public space. We propose that urban landscapes will contain "AI-parks" containing *trees of knowledge* that physicalize machine learning (ML) pathways that take on or augment the responsibilities of city departments and bureaus. The *trees of knowledge* are TUIs where humans can read and revise the inputs that civic AI systems learn from—an interaction that we call "pruning". *Topos* suggests that the interactions between AI systems and humans should be embodied and spatial in nature, so as to highlight the ways in which civic-oriented AI systems will directly affect the lived environments and multiple infrastructures of the urban landscape.

AI-Embedded Urbanism

Artificial intelligence (AI) is everywhere, and sooner rather than later, it will control a city near you. When a city's infrastructure is embedded with autonomous AI systems that can track pedestrians on the sidewalk, redirect driverless cars, and predict the rate of gentrification, the machine-readable city will become increasingly illegible and inaccessible to humans. *Could the city know itself better than you—the citizen—could ever know it*?

This question is at the center of *Topos*, which addresses the possibility of ubiquitous AI systems, and anticipates the shadows they might throw on the urban landscape. If the inner-workings of AI systems that drive the city are neither visible nor tangible, then how we design AI interfaces can illuminate the algorithmic dimension of the city for the people living in it. *Topos* imagines that the hidden intelligent systems controlling the city are made open and legible to citizens in the form of physical, manipulable, tree-like structures. How might *pruning* and *tending* these civic interfaces—these *trees of knowledge*—literally and figuratively reshape the urban landscape?

Designing AI into Public Spaces

We propose a new *typology*[1] (Steuteville, 2006) of public space that combines the mechanical qualities of urban dashboards with the permeable and spatial qualities of public parks. These new "AI-parks" contain *trees of knowledge* that physicalize what is otherwise invisible to citizens: the algorithms, decision trees, and neural nets that have taken on the responsibilities of city departments and bureaus.

AI-parks are maintained by civic workers, who tend, prune, and shape *trees of knowledge* so that the AI-embedded city can reflect public interests and, ideally, the public good. *Trees of knowledge* provide public insight to how civic data is being transformed by the various AI dimensions of the city.

If AI-parks are where civic affairs are conducted in plain sight and in real time, then *trees of knowledge* are tangible user interfaces (TUIs) that form a relationship between civ-

[1] A term that we are borrowing from the fields of architecture and urban planning. A *typology* is commonly defined as an "ordered assembly of building types" (Steuteville, 2006). We are expanding this term to encompass buildings and other well-defined or bounded spaces (such as public parks). Each of these types has a unique function (the activities that a space can contain), configuration (the relationships of internal spaces within the larger building or space), and disposition (the way the function and configuration of one building or space interacts with the buildings or spaces around it) (Steuteville, 2006). As opposed to simply being a new "type" of space that already fits into an assembly of familiar public spaces, we characterize our proposed AI-parks as constituting a typology; each park will have its own set of functions, configurations, and dispositions that connect it to other AI-parks and the urban landscape at large.

ic AI systems and the humans they are meant to support. Through the TUI, citizens and civic workers are able to *read* and *revise* these AI systems by interacting with them through the physical environment.

Informed by Shannon Mattern's survey of city control systems, we determined that these *trees of knowledge* would function as AI-interfaces, civic symbols, and platforms of alternative governance. Although *trees of knowledge* do the same work as urban dashboards by "render[ing] a city's infrastructures visible and mak[ing] tangible...various hard-to-grasp aspects of urban quality-of-life," (Mattern, 2015) they are not slick graphical summaries of the variables and metrics that describe the city from one quantifiable moment to the next. *Trees of knowledge* have tangled branches and gnarled roots—they are complex and non-reductive interfaces that embody what Mattern would call "dirty (un-'cleaned') data" (Mattern, 2015).

As a project, *Topos* is still answering the question of what these branching interfaces actually look and feel like, but moving away from reductive screen representations and toward more information-rich physical forms is central to our design motivations.

Design no.1: Experiential and Spatial Prototype

In order to play out our proposed human-to-AI interaction, we created an experiential prototype that simulated the reading and revision experience that our *trees of knowledge* would enable. Because AI systems don't naturally lend themselves to a tangible form, we confronted the reality that it is impossible to capture something as complex as a neural net in a determinate form.

As seen in figure 1, our first prototypes for the *trees of knowledge* were civic monument-scaled forms that manually contract and expand. The outer faces of this form serve as input layers and output layers for the neural nets that learn from city data in different ways; by unfolding and expanding the form, citizens and civic workers can respectively read and revise the hidden layers—where AI systems transform city data into intelligence.

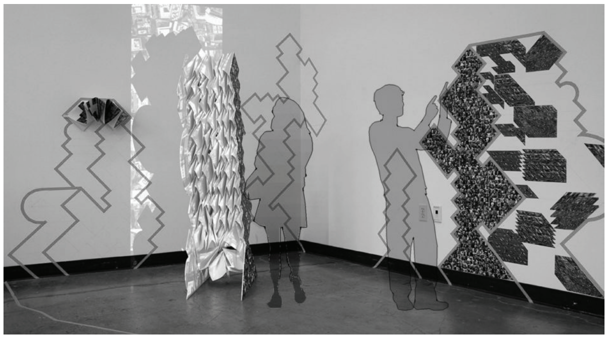


Figure 1. Installation view of experiential prototypes for citizen interaction with trees of knowledge.

Editable *trees of knowledge* enable citizens to add, subtract, emphasize, or de-emphasize elements on input layers in order to create different short- or long-term outcomes on civic matters—ranging from self-driving car congestion to urban green space development. Civic workers take these citizen annotations into account as they modify learning pathways in the hidden layers.

Design no.2: Visual and Animated Prototype

In the second iteration of *Topos*, we focused our energies not on what the human-to-AI interaction itself would look like, but rather how the act of "pruning" these *trees of knowledge* would affect the urban landscape at large.

In addition to producing an illustration that allowed us to rethink and redesign the *trees of knowledge* physically and structurally, we illustrated what a city full of AI-parks and their corresponding *trees of knowledge* might look like (see figure 2). To understand how these representations would lend themselves to our proposed human-to-AI interaction, we developed a short animated video wherein an anonymous citizen prunes a tree of knowledge, and the effects of their inputs to the physicalized AI system are simulated in an abstracted cityscape.

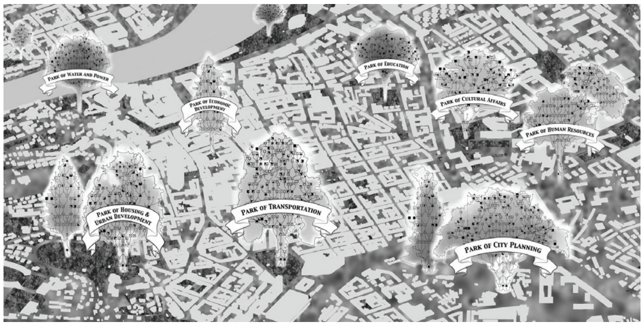


Figure 2. Illustration of what an urban landscape full of AI-parks and their corresponding trees of knowledge might look like.

AI and the "Right to the City"

At its core, *Topos* envisions a model of AI-embedded urbanism that guarantees what Henri Lefebvre calls the "right to the city"—an idea and social movement that advocates for the participation of individual and collective agents alike to shape the city (Lefebvre, 1996). It is in this image that cities have already attempted to model initiatives to formalize and concretize "the right to the city" for their inhabitants.

Knowing that an urban landscape unmodified by AI systems can ultimately sacrifice "the right to the city" to the hyper-present demands of privatization and capital, it is imperative for interaction designers to take on the problem of designing civic AI interfaces that will not allow this human right to recede into the shadows. It is easy for AI systems to reproduce more of the world we already have, but it will be up to designers to bring the complexities and contradictions of human-to-city interaction to the surface. When the intelligence of city government is outsourced to or augmented by artificial intelligence, designers must ensure that all citizens are still guaranteed the right to reshape their cities *in collaboration with* AI systems.

Taking *Topos* into The Real World

We want to take this symposium as an opportunity to bring our speculative design proposition about creating trust between humans and AI systems into conversation with more practical applications of interaction design and user experience strategy for AI. If *Topos* were implemented in a real world context, we acknowledge that our proposal of giving citizens their "right to the city" by interacting with *trees of knowledge* could go very wrong, very quickly. For instance, a citizen could tend to or prune the *trees of knowledge* in a damaging way—adding or taking away inputs that disrupt an existing balance of function, configuration, and disposition (Steuteville, 2006) between the AI-park and its outputs in the larger urban landscape.

By speaking to machine learning experts, we hope to learn more about how an AI system might actually deal with such damaging inputs—and therefore strengthen the argument that we are trying to make through *Topos*. How do AI systems "learn" from inputs that come later and are not part of the initial training data? Could we potentially design a "defense mechanism" into the *trees of knowledge* in order to regulate or prioritize inputs from citizens according to a greater sense of public good? If AI-parks put citizens in touch with AI systems (which are in themselves an extension of governance) through *trees of knowledge*, how can our TUI model the ability to both revise and *harmoniously negotiate with* an AI system and all of its existing inputs? Provided with answers to these questions, we hope to better connect our speculative vision of human-to-AI interaction in *Topos* with emerging real-world applications of AI in civic contexts.

Acknowledgements

Topos was originally developed in "The Internet of Enlightened Things", a class devised and mentored by Ben Hooker and Philip van Allen in the Media Design Practices program at ArtCenter College of Design in the Spring of 2017. In May 2017, *Topos* was presented at the Post-Internet Cities conference at the Museum of Art, Architecture, and Technology (MAAT) in Lisbon, Portugal. In September 2017, *Topos* was exhibited at Ars Electronica under the umbrella of "The Internet of Enlightened Things: AI in the Neighborhood".

About the Designers

Xiaoxuan (Sally) Liu is a designer passionate about using new media and technology to investigate the future of nature and urban spaces. Prior to ArtCenter, she studied Communication and Psychology at Syracuse University. **Godiva Veliganilao Reisenbichler** is a designer and artist who produces critical knowledge through visual, interactive, spatial, and written media. At ArtCenter, she investigates how technological interfaces form relationships between people, their mobile devices, and physical space. Godiva received a BFA in Painting and Art History from Washington University in St. Louis.

References

Lefebvre, Henri. 1968. *Writings on Cities*. Edited by Eleonore Kaufman and Elizabeth Lebas. Oxford: Blackwell.

Mattern, Shannon. 2015. Mission Control: A History of the Urban Dashboard. *Places Journal*. https://placesjournal.org/article/mission-control-a-history-of-the-urban-dashboard/.

Steuteville, Robert. 2006. Typology and urbanism: The disposition of types. *CNU: Congress for the New Urbanism*. https://www.cnu.org/publicsquare/typology-and-urbanism-disposition-types.

The UX of AI: Using Google Clips to Understand How a Human-Centered Design Process Elevates Artificial Intelligence

Josh Lovejoy
Google
lovejoy@google.com

Abstract

Google Clips is an intelligent camera designed to capture candid moments of familiar people and pets. It uses completely on-device machine intelligence to learn to only focus on the people you spend time with, as well as to understand what makes for a beautiful and memorable photograph. Using Google Clips as a case study, we'll walk through the core takeaways after three years of building the on-device models, industrial design, and user interface—including what it means in practice to take a human-centered approach to designing an AI-powered product.

Google Clips is a small form-factor camera that operates entirely offline using on-device AI. It can be stood up, held, or clipped onto things to capture candid photos of familiar people and pets.

Human-centered machine learning

As was the case with the mobile revolution, and the web before that, machine learning will cause us to rethink, restructure, and reconsider what's possible in virtually every experience we build. In the Google User Experience (UX) community, we've started an effort called "human-centered machine learning" to help focus and guide that conversation. Using this lens, we look across products to see how machine learning (ML) can stay grounded in human needs while solving for them—in ways that are uniquely possible through ML. Our team at Google works across the company to bring UXers up to speed on core ML concepts, understand how to best integrate ML into the UX utility belt, and ensure we're building ML and AI in inclusive ways.

Note that in this article, I will refer to ML as the process of training models and AI as the system architecture.

Just getting more UXers assigned to projects that use ML won't be enough. It'll be essential that they understand certain core ML concepts, unpack preconceptions about AI and its capabilities, and align around best-practices for building and maintaining trust. Every stage in the ML lifecycle is ripe for innovation, from determining which models will be useful to build, to data collection, to annotation, to novel forms of prototyping and testing.

We developed the following "truths" as anchors for why it's so important to take a human-centered approach to building products and systems powered by ML:

- Machine learning won't figure out what problems to solve. If you aren't aligned with a human need, you're just going to build a very powerful system to address a very small—or perhaps nonexistent—problem.
- If the goals of an AI system are opaque, and the user's understanding of their role in calibrating that system are unclear, they will develop a mental model that suits their folk theories about AI, and their trust will be affected.
- In order to thrive, machine learning must become multidisciplinary. It's as much–if not more so—a social systems challenge as it's a technical one. Machine learning is the science of making predictions based on patterns and relationships that've been automatically discovered in data. The job of an ML model is to figure out just how *wrong* it can be about the importance of those patterns in order to be as right as possible as often as possible. But it doesn't perform this task alone. Every facet of ML is fueled and mediated by human judgement; from the idea to develop a model in the first place, to the sources of

data chosen to train from, to the sample data itself and the methods and labels used to describe it, all the way to the success criteria for the aforementioned wrongness and rightness. Suffice to say, the UX axiom “you are not the user” is more important than ever.

Three ways human-centered design elevates AI

Addressing a real human need

This year, people will take about a trillion photos[1], and for many of us, that means a digital photo gallery filled with images that we won’t actually look at. This is especially true with new parents, whose day-to-day experience is full of firsts. During moments that can feel precious and fleeting, users are drawn to their smartphone cameras in hopes of capturing and preserving memories for their future selves. As a result, they often end up viewing the world through a tiny screen instead of interacting using all their senses.

What if we could build a product that helped us be more in-the-moment with the people we care about? What if we could actually be in the photos, instead of always behind the camera? What if we could go back in time and take the photographs we *would* have taken, without having had to stop, take out a phone, swipe open the camera, compose the shot, and disrupt the moment? And, what if we could have a photographer by our side to capture more of those authentic and genuine moments of life, such as my child’s *real* smile? Those moments which often feel impossible to capture even if one is always behind the camera. That’s what we set out to build.

Clips allows you to select the perfect frame and save it as a still. In this instance, I clipped the camera onto a basketball hoop to capture the moment just before my son made a basket (middle).

Guiding the intelligence

When we started the process, the most pressing question was: if people take tons of photos but don’t actually want to go back and curate them, how will we label ground truth? This is where the foundational “HCML exercise” was born: Describe the way a theoretical human “expert” might perform the task today. The theory was twofold: First, if a human can’t perform the task, then neither can an AI; second, by diving deep into the methods of an expert, we can find signal-to-guide data collection, labeling, and component model architecture.

The closest approximation I could think of was a wedding photographer, so I set out to find and hire contractors using a sufficiently ambiguous job posting. The interviews discussions were wide-ranging, but primarily centered on process. I wanted to find people who were particularly adept at deconstructing the many tiny decision-forks they employed in their craft We ended up discovering—through trial and error and a healthy dose of luck—a treasure trove of expertise in the form of a documentary filmmaker, a photojournalist, and a fine arts photographer. Together, we began gathering footage from people on the team and trying to answer the question, “What makes a memorable moment?”

It's important for us to recognize the amount of nuance, aesthetic instincts, and personal history that we often take for granted when evaluating the quality of our photos and videos. For example, I crack up every time I watch my younger son exploring the subtleties of a twisty straw (far left) or trying to juke my kisses (middle). And I well up with pride when I watch my older son on his bike at the park (far right), because I remember that day as a turning point in his self-confidence to ride on his own.

Building trust

The starting point for our work was an assumption that we could ‘show’ the model the stuff we thought was beautiful and interesting, and it would just *learn* how to find more. We had romanticized conversations about depth of field, rule of thirds, dramatic lighting, match cuts, and storytelling. But what I learned was that we should never underestimate the profound human capability to wield common sense; to quickly evaluate and prune the characteristics that are lacking in practical value.

These early experiments exposed crucial technical and methodological gaps that helped us reassess our assumptions about what the product could realize, as well as take stock in the unprecedented nature of the work. The reality of hand-held or body-worn video is that most of it is shaky, boring, poorly framed, or all of the above. So we shifted our paradigm from expecting ML to discover the most salient patterns—early models were actually quite fixated on things like hands close to the camera and abstract geo-

[1] InfoTrends Worldwide Consumer Photos Captured and Stored, 2013 – 2017

metric shapes—to understanding that it can only learn effectively under quite reductionist framings. Basically, we were trying to teach English to a two-year-old by reading Shakespeare instead of *Go, Dog. Go!*. This was where the myth of the AI 'monolith' crashed hardest for me; the idea that there's some singular 'intelligence' that understands all things and can generalize and transfer knowledge from context to context. We needed to reset our expectations and approach the task with far more pedagogy.

Back to basics

Consistency is the name of the game when trying to teach anything. It's why we wait as long as possible to unleash the madness of O-U-G-H (e.g. tough, through, thorough) on children when teaching them how to read and speak English. Spelling and pronouncing words like cat, bat, and sat, with their predictable "at" sounds, is so much more consistent!

With consistency comes confidence. Think about how quick—and eager—most students are to point out incongruity when a teacher provides two examples that don't seem to line up. Algorithms provide no such feedback. As far as an algorithm is concerned, everything they're shown is of equal value unless directed otherwise. For Clips, that meant we not only needed consistency between examples, but also within each example. Every individual frame needed to be representative of the specific prediction we're trying to teach it to make. And often that can come in the form of teaching it what to ignore.

Capture

We needed to train models on what bad looked like: hands in front of the camera, quick and shaky movements, blurriness.

We used examples like the above to train machine learning models to recognize when the camera was inside a pocket or purse (above, left), or when a finger or hand was in front of the lens (above, right). While it wasn't immediately intuitive to train models to ignore things, over time it became a crucial strategic piece in our design. By ruling out the stuff the camera wouldn't need to waste energy processing (because no one would find value in it), the overall baseline quality of captured clips rose significantly.

Composition

We needed to train models about stability, sharpness, and framing. Without careful attention, a face detection model will appreciate a face at the edge of the frame just as much as one in the center.

In an effort to train a model about subject continuity, it was important to selectively highlight examples where a subject was consistently well-framed (such as above, left).

Social norms

Familiarity is such a cornerstone of photography. You point a camera at someone and they offer implicit consent by smiling or posing. Moreover, you're the one looking through the viewfinder framing and composing the shot. With an autonomous camera, we had to be extremely clear on who is *actually* familiar to you based on social cues like the amount of time spent with them and how consistently they've been in the frame.

Editing

Diversity and redundancy is something we take for granted in the way we shoot photos; there's a little voice in the back of our head saying, "You haven't seen anything like this!" Or, "You've got enough shots of your kid for now, relax." But our models needed a lot of help.

We approached diversity along three different vectors:

- **Time**: The simple value of time passing is an important signal to appreciate. Don't go too long without capturing something.
- **Visual**: Subtle or dramatic changes in color can tell a lot about changes in environment and activity. Try to capture moments that have distinct aesthetic qualities.
- **People**: Are you in a big group or a small group or alone? Understanding how many different familiar faces you're encountering is a crucial part of feeling like you haven't missed important moments.

Trust and self-efficacy

One of the reasons we invested in Clips was because of how deeply important it was to demonstrate the importance of on-device and privacy-preserving machine learning to the world—not to mention its remarkable capabilities (e.g. it uses less power, which means devices don't get as hot, and the processing can happen quickly and reliably without needing an internet connection). A camera is a very personal object, and we've worked hard to ensure it—the hardware, the intelligence, and the content—ultimately belongs to you and you alone. Which is why everything—and I mean everything—stays on the camera until the user says otherwise.

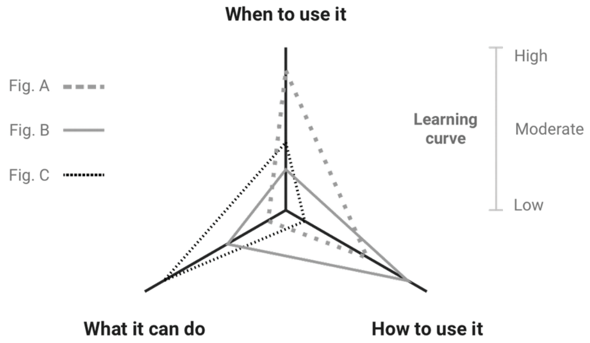
Concept budgeting

With an eye on trust and self-efficacy, we were also very intentional in the way we approached UI design. At the start of the project, that meant working through a few of our own funny assumptions about how "out-there" an AI-powered product needed to be.

When we reach into our brains for future-tech reference points, many designers will jump to the types of immersive experiences seen in movies like *Minority Report* and *Blade Runner*. But just imagine of how crazy it'd be to actually explain something like the UI in Minority Report to users: *Here, just extend your arm out, wait two seconds, grasp at thin air, then fling wildly to the right while rotating your hand counter-clockwise. It's easy!* Almost every sci-fi faux UI is guilty of something similar; as if the complexity of an interaction model needs to keep pace with the complexity of the system it's driving. But that's sort of where we were for awhile during our early design phase, and we got away with it in large part for three reasons:

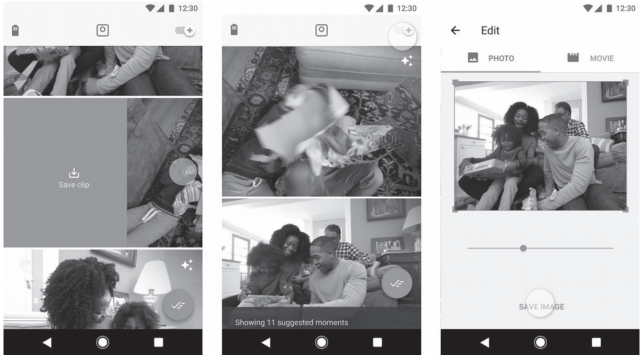
- We were showing people fake content in an obviously simulated environment, where they had no real connection to the imagery. Note that this issue isn't unique to AI; it's often one of the confounding factors when you bring people into the usability lab.
- We were surrounded by people every day who were all speaking the same language; thinking deep thoughts about AI-enabled futures. We were making the mistake of losing touch with the reference points that everyone else would bring to the table.
- We thought our new designs were super cool, so we gave ourselves a healthy amount of forgiveness when people didn't immediately get it.

Over time, we snapped out of it. We began fiercely reducing complexity in the UI, and made *control* and *familiarity* cornerstones of our experiential framework. We added a software viewfinder and a hardware capture button to the camera. We made sure that the user had the final say in curation; from the best still frame within a clip to its ideal duration. And we showed users more moments than what we necessarily thought was *just right*, because by allowing them to look a bit below the 'water line' and delete stuff they didn't want, they actually developed a better understanding of what the camera was looking for, as well as what they could confidently expect it to capture in the future.



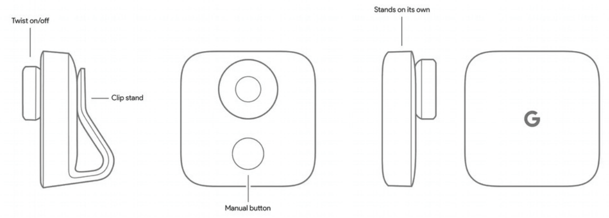
Most products have at least some learning curve, but with the added overhead of AI hype, it's especially important to 'spend' wisely on your user's cognitive load. When the context of use is novel to the user [figure A], bias for dependability. When there are a lot of new UI tricks to learn [figure B], make sure the primary use cases are super relatable. And when the functionality of the product is especially dynamic [figure C] , your UI should be flush with familiar patterns.

Through this process we discovered another critically important finding for testing an AI-powered product: fake it till you make it. If forced to choose, it's leaps-and-bounds more useful to prototype your UX with a user's real content using a Wizard of Oz approach than it is to test with real ML models. The latter takes an incredibly long time to build and instrument (and is far less agile or adaptive than traditional software development, so it's more costly to swing and miss), while the former affords you genuine insights into the way people will derive value and utility from your (theoretical) product.



Users preview their clips by streaming them from the camera. On the far left, users choose which clips they want saved to their phone. In the middle, users can toggle on a "suggested" view. On the right, users can pinpoint the exact frame they want to save as a still photo.

In the context of subjectivity and personalization, perfection simply isn't possible, and it really shouldn't even be a goal. Unlike traditional software development, ML systems will never be "bug-free"—insofar as a bug is defined as something that prevents the user from arriving at a specific linear outcome—because prediction is an innately fuzzy science. But it's precisely this fuzziness that makes ML so useful! It's what helps us craft dramatically more robust and dynamic 'if' statements, where we can design something to the effect of "when something looks sort of like x, do y." And in that departure from rigid logic rules, we also needed to depart from traditional forms of measuring engagement. Success with Clips isn't just about keeps, deletes, clicks, and edits (though those are important), it's about authorship, co-learning, and adaptation over time. We really hope users go out and play with it.



The camera turns on and off with simple twist of the lens and has a shutter button on the front for manual capture.

Designing with purpose

By re-orienting the conventional AI paradigm from finding ways to make the machine smarter, to exploring ways to augment human capability, we can unlock far greater potential in machine learning. It can become a tool for unprecedented exploration and innovation; a tool to help us seek out patterns in ourselves and the world around us. As human-centered practitioners, we have a tremendous opportunity to shape a more humanist and inclusive world in concert with AI, and it starts by remembering our roots: finding and addressing human needs through observation and experimentation, upholding humane values, and designing for augmentation[2], not automation.

The role of AI shouldn't be to find the needle in the haystack for us, but to show us how much hay it can clear so we can better see the needle ourselves.

[2] Augmenting Human Intellect: A Conceptual Framework, Engelbart 1962

FutureCrafting: A Speculative Method for an Imaginative AI

Betti Marenko
Central Saint Martins, University of the Arts London
b.marenko@csm.arts.ac.uk

Abstract

The issue I explore with this position paper concerns dominant cultural scripts around Artificial Intelligence (AI) and the need to imagine different narratives in light of machine learning's autonomous performativity. The aim is to offer a philosophical reflection, not only to sidestep narratives of techno-determinism, dystopia and existential risk to mankind, but also to speculate on how to imagine a (more) benevolent AI based on uncertainty and the co-evolution of humans and technology. The paper presents the speculative methodology I call *FutureCrafting*: a forensic, diagnostic and divinatory method that investigates the possibility of other discourses, equally powerful in building reality, constructing futures and having tangible impact. *FutureCrafting* is speculation at the juncture of design and philosophy, pivoting around the open-ended figuration of the *what if...?* It articulates collaboration rather than competition, coevolution rather than antagonism, and privileges the indeterminate and the imaginative. To conclude, the paper makes reference to the non-human intelligence of the octopus and to how this can inform a more imaginative AI.

Algorithm Narratives

As *the* cultural object of our present, the algorithm foregrounds a dominant techno-deterministic narrative that portrays computation as an almost mystical notion (Finn 2017) or even as a theocracy (Bogost 2015). In such a narrative, rationalization and logic coexist with deep myth – the ancestral belief in invisible forces. On one hand, we, users/content providers, like to believe that algorithms are efficient, logical, and clean procedures (they are not). On the other, we embrace a faith-based approach, the same conviction that ancient seekers would have had in the murmuring of an oracle.

Algorithms create reality in ways that are both alluring and evident, opaque and controlling. We use them "as pieces of quotidian technical magic" (Finn 2017, 16). We trust them with our many choices - partners, music, books; we are given or denied credit, job, insurance; we are fed tailored search results and social media updates. And yet, we hardly understand how they work; indeed, not even the programmers know. The simplistic notion of algorithms as procedural problem-solving entities, i.e. what turns questions into answers (according to Google) does no longer suffice. In particular, it cannot account for the uncertainty growing at the core of computation (Parisi 2013, 2017). New narratives are needed, that can turn uncertainty into an asset rather than reducing its ambiguity and providing explanations that rely solely on human-centered models.

AI Speculation

The importance of speculation emerges when we consider that Machine Learning's (ML) way of working is highly inductive, unlike traditional deductive AI approaches. ML starts from real observable behaviors expressed and captured in the the form of data. From here, verifiable models of given behaviors are built; a range of tasks (clustering, classifying, categorizing, matching) is performed; then, similar future behaviors are predicted.

With ML performing a continuous automatic revision and refinement of models based on a constant supply of fresh data, we enter a *meta-digital* phase (Parisi 2017), where new levels in the automation of registration, mobilization and communication are taking place. As the operative mode of AI shifts from validation to discovery through inductive data-retrieval and recursive training, at the core of this process we find uncertainty, indeterminacy, and unknowns. When the machine no longer simply searches for information but combines and recombines data to train itself, contingency enters the process and must be accounted for. This has profound implications on current AI narratives, and it must inform how to imagine and conceptualize near future AI.

Digital theorist Luciana Parisi (2013; 2017) argues that if AI is rooted on uncertainty, then it must be understood as a non-conscious form of cognition, possessing its own non-human way of learning. To clarify: this does not mean to advocate an overbearing machine rationality antagonist to humankind. Rather, it means to acknowledge that *what* machines can do does not coincide necessarily with *how* they think. What is needed, then, is a speculative critique of ML, inspired by abductive reason - the formulation of interrogative hypotheses (such as *what if…?*) - and finely attuned to the contingent, the unpredictable and the uncertain (Marenko 2015). This is speculation in action – *FutureCrafting* – a method that prioritizes imagination over direct observation, and that aims at capitalizing on the indeterminate. Speculative approaches to design (Dunne and Raby 2014) and the field of 'design fiction' (Coles 2016) have shown how to deploy design to suggest alternatives to the existent, ranging from the possible to the implausible, so to provoke debate, critique and reflection. Though FutureCrafting resonates with (and stems from) similar concerns and is likewise engaged with expanding the remit of what design *can* do, it puts however greater emphasis on the theoretical framework that supports its methods. Acknowledging a legacy of philosophical ideas, concepts and discourses is a crucial aspect of FutureCrafting, one that both grounds and propels forward its endeavor. The practice of contesting received notions of technology, inventing new modes of human-machine interaction, and speculating on different futures, cannot be disjoined from the risky business of operating at the edge of thinking. Here is where the power of the imagination in seizing alternative possibilities becomes a radical tool for change and acquires political valence. The challenge then would be: how to exploit the potential of digital uncertainty in ways that feed into new collaborative models of human-machine interaction? (Marenko and Van Allen 2016).

The Robot Does Not Exist

French philosopher and technologist Gilbert Simondon's work is illuminating here (2017). His notion of technogenesis, that is, the evolution of technical objects, is based on the idea of the co-habitation of humans and technology. Technical objects, including algorithms and AI, are always the temporary concrete expression of a morphological spontaneous evolution, which depends neither on natural processes nor on human design exclusively. Far from evolving in isolation, technical objects are the result of a process where internal parts converge and adapt according to a principle of internal resonance. This process *(concretization)* describes a coming together of functions by which the object acquires an internal coherence that propels it beyond the intention of its inventor. Even though they are designed and made by human beings, then, technical objects have a life of their own (Schmidgen 2012).

This argument is important for two reasons. First, it provides an epistemological shift that fully integrates technology into culture. The boundary between the natural and the artificial, the animate and the inanimate, the human and the non-human becomes blurred. Put differently, we can say that humans are always already among machines and, more broadly, among everything that is not human. Likewise, technical objects and, more broadly, everything that is not human, are always already among, and co-evolving with, humans. The second implication of Simondon's technogenesis is that it helps us frame and understand how technical objects, as they evolve, acquire autonomy – a valuable insight to use to conceptualize ML and to speculate imaginatively on AI. Indeed, this means something else too: that to talk about 'artificial' intelligence is incorrect. There is only *one* intelligence, constantly morphing and evolving. Perhaps this is the real meaning of what Simondon wrote in 1958: "The robot does not exist".

Conscious Exotica

But how can we exercise our human imagination to speculate on alternative AI narratives? An interesting viewpoint is presented by computer scientist Murray Shanahan who poses provocative questions concerning what he calls 'the space of possible minds' where humans could encounter radically alien and exotic forms of cognition (2016). By stating that "there's no reason to suppose that a human's capacity for consciousness could not be exceeded by some other beings", he takes the reader on an imaginative journey exploring this possibility.

What matters greatly is the method. In describing his experiment as "fanciful", Shanaham shines a light on the significance of adopting a speculative frame of inquiry when dealing with AI's uncharted territories. He positions a number of diverse human and non-human entities on a diagram whose two axes maps human likeness (H-axis), and capacity for consciousness (C-axis). A creature like the octopus, for instance, scores high on the C-axis (it is cognitively sophisticated), but low on the H-axis (it is quite hard to comprehend from our human perspective).

"The most exotic sort of entity would be one that was wholly inscrutable, which is to say it would *be beyond the reach of anthropology*" (Shanahan 2016). In other words, humans would need to think in radically non-anthropocentric ways, even reappraising what human consciousness is. It may be, continues Shananan, that a shift is required, from a monolithic notion of consciousness – made of memory, world and self awareness, capacity for empathy, emotional and cognitive integration - to a disaggregated, more distributed form of consciousness. To successfully speculate on imaginative AI, then, one route is to bypass the need to mimic human biology and to look instead at what non-human intelligences have to offer.

Cephalopod Cognition

Recent research on cephalopods, and the octopus in particular, show that these creatures may be specialists in distributed control systems (Grasso 2014, Godfrey-Smith 2016). Some types of octopuses (the common octopus *Octopus vulgaris*), possess fewer neurons in the brain than in the peripheral nervous system. With two thirds of its neurons located in the arms, the octopus has effectively two brains. Its neural system is exceptionally decentralized. Its arms are autonomous agents. Thanks to such a decentralized information processing system, the octopus can provide an innovative perspective on neural architecture and efficient distributed cognition (Laschi 2016). The octopus's brain does not issue top-down commands for every small movement of the arms. While the brain initiates motion, the lower motor centers control the precise neuromuscular activity. Experiments have shown that a severed arm will continue to act, search for food, and once found, it will bring it to the place where the mouth is supposed to be. Even more remarkably, the octopus' limbs do not need comprehensive direction to produce the desired movement, but respond to environmental stimuli in adaptive ways. Each one of the eight arms can be taken as a complex distributed information processing structure, able to act and problem-solve autonomously. For instance, while the octopus is busy checking a cave, a tentacle can be engaged with prodding a shellfish.

As a paradigmatic example of embodied and distributed cognition, it is no wonder that the octopus has become a model for soft robotics and AI research. This has led to the first entirely soft *octobot* recently developed by Harvard scientists (Burrows 2016). Also, inspired by the octopus' behavior, roboticist Alfonso Íñiguez (2017) has designed a system with a CPU that does not spend resources in micromanaging coprocessors, exactly like the octopus' central brain does not spend resources in micromanaging its arms. The potential of mimicking the complex neural system of the octopus is also studied by the U.S. defense contractor and industrial corporation Raytheon (2016), conducting robotics experiments with a network of machines that work together in a semi-autonomous way through coordination by a central command unit and a pack of independent agents. Applications are envisioned in the design of self-balancing biped robots thanks to the central brain's ability to delegate (Íñiguez 2017). There are parallels here with 'edge computing' - advanced on-device processing and analytics (Talluri 2017) where AI computation is pushed to the *edge* of the network (rather than the cloud) as close to the sensor/actuator as possible.

As perhaps the closest form of alien intelligence that we can study, the octopus is the blueprint for the development of an autonomous AI with neural networks that adapt to, and learn from, the environment. It could offer the seed of a new narrative rooted on non-human consciousness.

FutureCrafting

Scholarship at the intersection of design and sociology indicates the need to combine speculative design methods with humanities methodologies to capture social events that are ontologically open, processual and emergent (Michael 2012, Smith 2016). I would argue that AI's future narrative landscape demands a speculative approach. Expanding on this "inventive problem-making" (Michael 2012) FutureCrafting reconceptualises contingency and rethinks uncertainty by treating them both as a material to work with, rather than as a risk or a threat to avoid.

FutureCrafting gives shape to the future, and does so here and now. *Future* is about speculating, but avoiding the trap of escaping into a fantasy of what the future could or should be. Instead, FutureCrafting captures the future, grabs it and brings it back to the here and now, so to inform the present. Which is the *Crafting* part: crafting pertains exquisitely to the now. FutureCrafting is speculation by design, a performative rather than descriptive strategy, whose interventions are designed to prompt, probe, and problematize, to inject ambiguity and even the non-rational and the non-sensical. To borrow philosopher Isabelle Stengers' words on "speculative methodologies", FutureCrafting is a practice that "affirms the possible, that actively resists the plausible and the probable targeted by approaches that claim to be neutral" (Stengers 2010, 57).

Framed in this way, FutureCrafting is a strategy and a stratagem to conjure new figures of thought. It provides a set of tools at once forensic, diagnostic, and divinatory. It is *forensic* because it concerns things taken as witnesses so to articulate the existent. It is *diagnostic* because it invents explanatory hypotheses in an interrogative fashion – as said, it relies on abduction, and it is unconstrained by a priori theory or a posteriori verification. It is *divinatory*, because it attracts future images around which new thoughts can coalesce.

FutureCrafting gives priority to imagination over direct observation, searches for the least familiar hypotheses, those with no verifiable answer, and leans toward the production of what is not there yet. It is driven by the question *what if?* Precisely because it has affinity with practices bent on divining, predicting and conjuring, it is a fine instrument to probe what ML is doing today and will be doing tomorrow.

Bio

Betti Marenko is a design theorist, academic, and educator. She has a background in philosophy, sociology and cultural studies, and a decade of experience in design education. Her interdisciplinary approach brings together design studies, continental philosophy and the analysis of digital cultures to investigate the relationships between design, society and technologies, and their role in shaping possible futures. Betti's work features regularly in international conferences, collections and peer-reviewed journals such as *Design and Culture*, *Design Studies* and *Digital Creativity*. She is the co-editor of *Deleuze and Design* (Edinburgh University Press 2015, with Brassett) - the first book to use Deleuze and Guattari to provide a new theoretical framework for the theory and practice of design. She is Contextual Studies Programme Leader for Product Design, Central Saint Martins, University of the Arts London, UK.

Statement

I am currently writing a book titled *Digital Uncertainty. Between Prediction and Potential in Algorithmic Culture*, which investigates the new contingent logic of planetary computation and its impact on society, publics and subjectivities. The book looks at the effects of the growing autonomy and unpredictability of digital technologies, machine learning algorithms and AI. By connecting philosophy and computational theory to design, my research brings a holistic interdisciplinary approach to the issue of digital uncertainty and launches a debate on its unexplored potential. I am interested in bringing into dialogue AI developers, interaction and speculative designers, programmers and engineers, to provide new insights around digital experience, interrogate current theoretical positions and inform interdisciplinary debates on human-machine interaction. The symposium will offer this opportunity.

References

Bogost, I. 2015. The Cathedral of Computation. *The Atlantic*

https://www.theatlantic.com/technology/archive/2015/01/the-cathedral-of-computation/384300/

Burrows, L. 2016. The First Autonomous, Entirely Soft Robot. *Harvard Gazette*, 24 August.

http://news.harvard.edu/gazette/story/2016/08/the-first-autonomous-entirely-soft-robot/

Coles, A. ed. 2016. *Design Fiction*. Berlin: Sternberg Press.

Dunne, A. and Raby. F. 2014. *Speculative everything. Design, fiction and social dreaming*. Cambridge, Mass. and London: MIT Press.

Finn, E. 2017. *What Algorithms Want. Imagination in the Age of Computing*. Cambridge, Mass. and London: MIT Press.

Godfrey-Smith, P. 2016. *Other Minds. The Octopus and the Evolution of Intelligent Life*. London: Wlliam Collins.

Grasso, F. W. 2014. The Octopus with Two Brains: How are Distributed and Central Representations Integrated in the Octopus Central Nervous System? In Darmaillacq, A., Dickel, L., and Mather, J. eds. *Cephalopod Cognition*. Cambridge: Cambridge University Press. 94-122.

Íñiguez, A. 2017. The Octopus as a Model for Artificial Intelligence - A Multi-Agent Robotic Case Study. In Proceedings of the 9th International Conference on Agents and Artificial Intelligence, 2: ICAART, 439-444, Porto, Portugal.

http://www.scitepress.org/DigitalLibrary/PublicationsDetail.aspx?ID=QNu8OYOoE1c=&t=1

Laschi, C. 2016. Robot Octopus Points the Way to Soft Robotics With Eight Wiggly Arms. *IEEE Spectrum.*

https://spectrum.ieee.org/robotics/robotics-hardware/robot-octopus-points-the-way-to-soft-robotics-with-eight-wiggly-arms

Marenko, B and Van Allen, P. 2016. Animistic Design: How to Reimagine Digital Interaction between the Human and the Nonhuman. *Digital Creativity*. Special issue: Post-anthropocentric Creativity. Stanislav Roudavski and Jon McCormack eds. London: Routledge. 52-70.

Marenko, B. 2015. When Making becomes Divination: Uncertainty and Contingency in Computational Glitch-Events. *Design Studies* 41. Special issue: Computational Making. Terry Knight and Theodora Vardoulli eds. London: Elsevier: 110-125.

Michael, M. 2012. De-signing the Object of Sociology: Toward an 'Idiotic' Methodology. *The Sociological Review*, 60(S1):166-183.

Parisi, L. 2017. Reprogramming Decisionism. *e-flux* 85 www.e-flux.com/journal/85/155472/reprogramming-decisionism/

Parisi, L. 2013. *Contagious Architecture.* Cambridge, Mass. and London: MIT Press.

Raytheon 2016. Synthetic Smarts. With Learning Robots and Emotional Computers, Artificial Intelligence becomes Real. www.raytheon.com/news/feature/artificial_intelligence.html

Schmidgen, H. 2012. Inside the Black Box: Simondon's Politics of Technology. *SubStance* 41(3,129. Madison: University of Wisconsin Press. 16-31.

Shanahan, M. 2016. Conscious Exotica. *Aeon*

https://aeon.co/essays/beyond-humans-what-other-kinds-of-minds-might-be-out-there

Simondon, G. 2017. *On the Mode of Existence of Technical Objects*. Minneapolis: Univocal.

Smith, R.C. et al. eds. 2016. *Design Anthropological Futures*. London: Bloomsbury.

Stengers, I. 2010. *Cosmopolitics I*. Minneapolis: University of Minnesota Press.

Talluri, R. 2017. Why Edge Computing is Critical for the IoT. *NetworkWorld.* 24 October https://www.networkworld.com/article/3234708/internet-of-things/why-edge-computing-is-critical-for-the-iot.html

A Panel on Cybernetics and the User Experience of AI Systems

Nikolas Martelaro
Stanford University
424 Panama Mall, Bldg. 560
Stanford, California 94305

Wendy Ju
Cornell Tech
2 W Loop Rd.
New York, NY 10044

Abstract

Cybernetics was influential in the early age of AI and might hold the keys towards making AI systems more interactive. Our panel will explore cybernetics as a useful framework for designers of artificially intelligent (AI) systems. Our four panelists—Hugh Dubberly, Deborah Forster, Jody Medich, and Paul Pangaro—will each discuss how they have used cybernetic theory in their own work. We will then delve into a discussion about the future design of AI systems and the areas where cybernetic theory may prove useful for user experience design.

Cybernetics and Artificial Intelligence (AI) have often been closely associated and even equated with each other, though they each take different approaches to understanding and developing intelligent systems (Papert 1988). While AI focuses on the creation of intelligent systems based on computers, cybernetics is broadly interested in understanding communication and control within interacting systems—systems which can be biological, mechanical, computational, or social (Wiener 1961). Moreover, cybernetics is interested in understanding and designing the interaction between intelligent systems and elevates *action* and *interaction* as a means of generating intelligent behavior (Pangaro 2006). By examining systems from the perspective of interaction that has been lost in the current-day AI boom, we believe we will discover a useful framework for the user experience design of any intelligent system.

New technological advances now allow for AI to be widely used in everyday products and services that interact with people. With these advances, designers should understand and equip themselves with tools for creating systems that can learn and adapt with their users to meet people's needs. Learning and adapting to the needs of a system are the goals of both designers and cybernetics (Dubberly and Pangaro 2015). It has been suggested, and we agree, that cybernetics is the "silent partner" of design (Glanville 1999), and provides a useful framework for assisting designers in creating intelligent human-centered systems (Krippendorff 2007; Dubberly and Pangaro 2015).

In this panel, we plan to discuss the topic of cybernetics in relation to the design of AI systems that interact with people. Our goal for the conversation will be to explore how cybernetics can be useful for the user experience design of AI systems. We will discuss how cybernetics has influenced the ways our panelists view the world and how it has shaped their own work. We will then devote the majority of our discussion to the ways that cybernetic theory can benefit designers in creating new intelligent systems. Some questions that we will explore include:

- What role will designers play when creating systems that learn on their own?
- What aspects of communication design are important for facilitating smooth user interaction with intelligent systems?
- How will designers and users control and update how systems behave?
- How can designers use and manage interactions with many intelligent systems distributed across the user's environment and the Internet?

Panelists

Throughout the years, cybernetics has championed itself as a way of understanding and making change in the world across many disciplines. It has brought together people and encouraged discussion from mathematicians, biologists, engineers, anthropologists, sociologists, designers, and economists. We look to bring together a group of people from different backgrounds within academia and industry to share their perspectives on cybernetics and design.

Hugh Dubberly is a design planner and teacher. At Apple Computer in the late 80s and early 90s, Hugh managed cross-functional design teams and later managed creative services for the entire company. While at Apple, he co-created a technology-forecast film called Knowledge Navigator, that presaged the appearance of the Internet in a portable digital device. While at Apple, he served at Art Center College of Design in Pasadena as the first and founding chairman of the computer graphics department. Intrigued by what the publishing industry would look like on the Internet, he next became Director of Interface Design for Times Mirror. This led him to Netscape where he became Vice President of Design and managed groups responsible for the design, engineering, and production of Netscapes Web portal. Hugh graduated from Rhode Island School of Design with a BFA in graphic design and earned an MFA in graphic design from Yale.

Deborah Forster is a primatologist and cognitive scientist, currently a research specialist at the Contextual Robotics Institute at UC San Diego. She studied social complexity and distributed cognition in olive baboons in Kenya, developing a state-space (and time series) approach to analyzing complex social behavior. Forster applied this relational systems framework in her work with car designers, intelligent driver support systems research, architecture education, social robotics research, art-science collaborations, and movement education practice. Her current projects support interdisciplinary design teams developing infant biometrics, automated pain detection in horses and other animals, cognitive robotics, and autonomous transportation research.

Jody Medich creates superhumans, not supercomputers. She uses perceptual computing (AI, machine learning, AR/VR, robotics, sensors, etc.) to make technology as easy to control as our own body and mind; creating tools that help humans become more powerful. Today, she is Director of Design for Singularity University Labs, where she incubates solutions to Global Grand Challenges using exponential technologies. Her previous work includes User Experience (UX) design for DARPAS Big Dog, Principal Experience Designer on Microsoft HoloLens, Principal UX at LEAP Motion, and UX Strategy for Toyota's AiCar. Jody is also a practicing artist with an MFA in Painting and Design + Technology from the San Francisco Art Institute.

Paul Pangaro is Chair and Associate Professor for MFA Interaction Design at the College for Creative Studies in Detroit. His career spans roles as teacher and curriculum designer; chief technology officer, product designer, and co-founder in tech startups; consultant in organizational effectiveness and innovation; and future-caster all from the perspective of cybernetics as a frame for understanding and designing systems for conversation. He holds a BS from MIT in Humanities/Computer Science and a Ph.D. from Brunel (UK) in Cybernetics where his dissertation advisor and then collaborator in government research contracts was Gordon Pask, founder of Conversation Theory.

Author Biographies

Nikolas Martelaro is a Ph.D. student in Mechanical Engineering at Stanford University's Center for Design Research DesignX Group. His current work focuses on how computationally-aware physical products can elicit meaningful interactions with users and how these products can relay those experiences back to designers.

Wendy Ju is an Assistant Professor of Information Science at Cornell Tech. Her current research in the areas of physical interaction design and ubiquitous computing investigates how implicit interactions can enable novel and natural interfaces through the intentional management of attention and initiative.

References

Dubberly, H., and Pangaro, P. 2015. Cybernetics and design: Conversations for action. *Cybernetics & Human Knowing* 22(2-3):73–82.

Glanville, R. 1999. Researching design and designing research. *Design issues* 15(2):80–91.

Krippendorff, K. 2007. The cybernetics of design and the design of cybernetics. *Kybernetes* 36(9/10):1381–1392.

Pangaro, P. 2006. Interaction – Cybernetics – Citroëns.

Papert, S. 1988. One ai or many? *Daedalus* 1–14.

Wiener, N. 1961. *Cybernetics or Control and Communication in the Animal and the Machine*, volume 25. MIT press.

The Design of the User Experience for Artificial Intelligence

Christine Meinders, Selwa Sweidan

Co-founders of Artificial Knowing, an AI Innovation Consultancy
1206 Maple Avenue, Suite 1032, Los Angeles, CA 90015
info@artificialknowing.com

Abstract

We share two prototypes that explore different aspects of the design and application of inclusive AI. This approach to inclusive AI Design seeks to engage typically excluded communities, such as individuals of varying socioeconomic status, race, age, gender (and those who do not identify with a gender), as well as to critique and explore alternatives to conventional AI Design.

Introduction

There are many ways to approach intelligence and many definitions of artificial intelligence. This paper uses Nils J. Nilsson's definition: "Artificial intelligence is that activity devoted to making machines intelligent, and intelligence is that quality that enables an entity to function appropriately and with foresight in its environment" (Nilsson 2010). Similarly, there are multiple ways to approach Artificial Intelligence (AI) Design. This paper presents an inclusive approach to Artificial Intelligence (AI) Design, which we frame as being part of a practice we call Knowledge Design. Referencing Alison Adam (Adam 1998), in this practice, knowledge encompasses the artificial life and intelligence spectrum, while at the same time honoring different ways of thinking and knowing. Thus, the process of AI Design we propose is collaborative and it defines the context of the "knowledge" upon which an entire (intelligent) system is structured. In other words, Knowledge Design allows for conversations about wanted and unwanted bias in AI systems, while also modeling an inclusive approach to authoring and sourcing contexts and data.

We see AI Design as a material practice of working with code and context (or sociocultural considerations) to frame and generate computational experience. In essence, this can be simply and reductively stated as AI Design = (code x material x context) + (experience x form). In this paper we combine development concepts with physical objects (such as products) and digital materials (code) to produce form and critically intelligent cultural interactions.

Under the umbrella of Knowledge Design, we present an approach to AI Design that is inclusive, embodied, and co-creative. In practice, this translates to collaboratively interrogating concepts (knowledge) with stakeholders, creating prototypes and bringing those prototypes to a community. We share two research projects, "Intelligent Protest" and "Accumulative Collaboration," which address the question of how we conduct AI Design from an inclusive perspective and how this approach generates conversation and co-creation with a range of communities not typically included in the design and implementation of AI systems (on excluded communities, see Byrnes 2016). Our process allows us to co-create and train data inclusively—with and for the community the intelligent system will serve. Further, these projects demonstrate an embodied approach to the creation of training data, which allows us to generate new conversations and insights, design for excluded communities, and explore models for training individually curated algorithms or systems trained by specific nontraditional user types.

Excluded communities, included bodies

The inclusive AI Design utilizes an embodied approach to conducting training that can generate unique data tied to a location or object. In our research we ask questions such as, what does it mean to use computer vision to allow access to buildings, parking garages, cars and apartments, and what are the social implications of purchasing products with pre-trained data sets over products that include all members of a community (and can be trained on small sample data)?

Overall, an embodied approach to AI Design offers two advantages. First, participants with limited exposure or understanding of intelligent systems encounter less of a barrier when they are able to engage with a system through their body. Instead of introducing linear regression in training a data set, for example, or relying on participants' computer literacy (which can be exclusionary), the participant interrogates the system through facial expressions or hand gestures. This empowers participants with any level of knowledge to engage with a system, and the form of engagement often looks and feels like play, which conveys to participants that there is no "correct" way of interrogating a system. This playful, embodied approach to co-creation and research allows for a very wide range of feedback and insights. In "Intelligent Protest," participants engaged with the system through their bodily presence and facial expressions, and in "Accumulative Collaboration" through simple hand gestures.

Second, an embodied approach to co-creation and training of data sets also reinforces inclusive design by designing with and for all bodies. Designing with different bodies from the outset can allow us to think about what it means to design across variances in hair, beard, skin, size, ability and so forth, especially in the digital space, not only to effectively design these products, but to reduce bias in things like auto tagging and image recognition. Although we need to approach AI Design from an inclusive perspective so these technologies can work on all bodies, we must also consider ways to guard against potential discernments from machine learning advances, such as algorithms that purportedly identify sexuality (Wang and Kosinski 2017; significantly, this paper is now under ethical review), and the ramifications of using such tools in conservative societies. An embodied, community-generated training data approach allows the AI Designer to decrease algorithmic bias, such as the other race effect (own race bias) evidenced in face recognition algorithms (Phillips, et al., 2011). Recognizing that human bias can be translated to bias evidence in algorithms, this embodied approach to co-creating with typically excluded communities allows the AI Designer to include and acknowledge multiple, diverse, and varied bodies and experiences.

Methodology

Dara Blumenthal's research proposes that living-sensory embodiment is an ongoing process, and looks at the body as beyond being enfleshed (Blumenthal 2014). Paul Dourish suggests that everyday human interaction is embodied (Dourish 2001), but while he highlights embodiment and offers guidelines, he refrains from offering a model or methods for embodied approaches to human-computer interaction (HCI). We apply this lens of embodiment to AI Design, updating "interactive system" to "intelligent system" in Dourish's argument, while additionally taking the step of sharing methods for engaging in an embodied research practice.

Embodied Approach, Different Data

Performative Prototyping (Sweidan 2016) is a proprietary method that harnesses movement-based research to prototype from an embodied perspective. Performative Prototyping updates HCI research methods to engage embodied thinking in the research process (specifically in the ideation and prototyping phases). The AI Designer leads the participant through an imagined scenario or a designed system which requires movement and physical engagement. Performative Prototyping intersects traditions of dance improvisation and somatic research with HCI. It draws from "critical making" (Ratto and Boler 2014) in that the act of prototyping is framed as a means of interrogating and unpacking the assumptions and conceptual framework of the designed artifact. Performative Prototyping also draws from qualitative research practices in the HCI space, such as the "think aloud" methodology (Lewis and Rieman 1993) which includes a debriefing process involving extensive questioning of the participant following the embodied action/enactment. Performative Prototyping is both divergent and affords a low barrier for participation since basic movements (such as walking) can be harnessed to allow workshop participants to engage in basic system design.

In practical terms, collaborative, embodied AI Design entails using AI systems and machine learning tools to encourage human-to-human and human-to-machine connections. Our research does not result in one finished product, but rather a collection of prototypes, designed for experiences in the AI Design space. These prototypes serve as tools that help us envision how to design for and with intelligent systems, allowing us to move outside of the product-driven design space into the inclusive, intelligent experiential space.

The two projects we present include the following methods:

- Co-creation and community research: we conducted research in various locations with different communities in Los Angeles. We intentionally sought to prototype with audiences that were varied in age, race/ethnicity, SES, gender (and non-gender), and technical background. We took special care to target audiences that were not primarily cis male. The project "Intelligent Protest," was a year-long research project in which we were invited to specific communities around Los Angeles. This was carefully curated so voices that are typically not heard in the AI Design space were a part of the co-making project. In the project "Accumulative Collaboration," we playfully explored what it means to use the physical bodies of artists as material for the training data, sourcing people as data.

- Performative Prototyping.
- Wekinator is an open source machine learning tool.

AI Design Research Projects

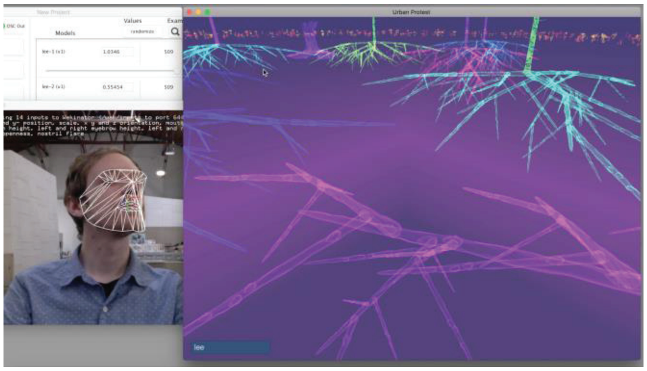
"Intelligent Protest" and "Accumulative Collaboration" are two experimental prototypes which utilize co-creative and embodied research methods and illustrate our vision of AI Design within a broader Knowledge Design practice. Both projects feature inclusive ways to think about different aspects of design and implementation. By engaging with typically excluded communities, we explore alternative explorations to conventional approaches to AI Design.

Intelligent Protest

The project "Intelligent Protest" stemmed from our collaborative AI research group, "Feminist AI Projects: Bits and Bytes." The research and design of this project involved a year of holding local community workshops that provided access to AI Design tools, with a particular outreach to those who have not been socialized to participate in shaping technology and its applications. A pilot series of AI workshops was planned to foster gender-equitable, creative tech spaces in which small working groups agreed upon a mutual area of concern (such as immigration reform). Then, drawing upon their collective skills, the group explored the potential of the AI tools to create a project around the area of concern. The groups consisted of students, mothers, software engineers, makers, researchers, and artists. This research resulted in new thinking and outcomes in the AI Design space and explored new experiences in civic engagement.

Using the Intelligent Protest prototype, individuals can login from a home computer and participate in the virtual protest space. Additionally, this virtual space can be utilized and displayed at an actual physical protest site, using AI Design and physical movement to bridge physical and virtual worlds. This application of embodied research with the community exemplifies broader thinking around what it means to embody artificial knowledge from a research and design perspective (Meinders 2017).

During the "Intelligent Protest" project, individuals used their bodies to engage in a collaborative protest in virtual and physical spaces. The embodied expression of protest emerged from the co-creators' desire to scream using new parts of (or the whole) body, not just a voice. This framed the way we prototyped our protest and allowed for multiple bodies to strengthen the experience of the protest. A virtual sit-in was created by using Rebecca Fiebrink's machine learning tool Wekinator with Open-Frameworks' detailed facial feature tracking software to occupy a virtual sit-in, and a collaboratively created app (using the game engine Unity), in response to protesting tree removal in the city of Alhambra, CA (Fiebrink 2009; Kogan 2015). When individual users launched Wekinator, the Unity app, and the facial feature tracking software, they could provide training examples of facial movements which were mapped to outputs in the Unity app. For example, when an individual's tree avatar roots connected with the roots of other trees, they acquired the sound associated with the other trees' roots. Users thus can be present and are rewarded the longer they are in the space, collecting the sounds of other avatars once the tree roots interconnect. Users' avatars remained for twenty-four hours. The idea of using body information (biometrics, facial recognition) in civic discourse makes it possible for individuals working multiple jobs, or caring for children and parents, to participate in civic engagement.



```
ng 14 inputs to Wekinator [illegible] to port 64
nd y- position, scale, x y and z orientation, mout
 height, left and right eyebrow height, left and
penness, nostril flare
```

Users engage with Wekinator to connect with other protesters in an avatar sit-in.

To coordinate this sit-in, we set up Wekinator to receive 14 input values and compute 5 continuous output values which were mapped to an avatar in the Unity game engine. We selected Wekinator's default neural network algorithm and used 5 collaboratively designed facial movements to train the neural network for the face protest. These outputs were used in a designed Unity environment, where each individual who logged in had an avatar of a tree with roots. The roots were created by a simulation of a Lindenmayer System (L-System) and the 5 outputs affecting the individual avatars in the collaborative protest space were:

Output 1. Rotation of tree canopy
Output 2. Modified root color (constrained to hues near the hue of the canopy)
Output 3. Root network growth rate
Output 4. Level of audio distortion
Output 5. Cut-off frequency for audio low-pass filter

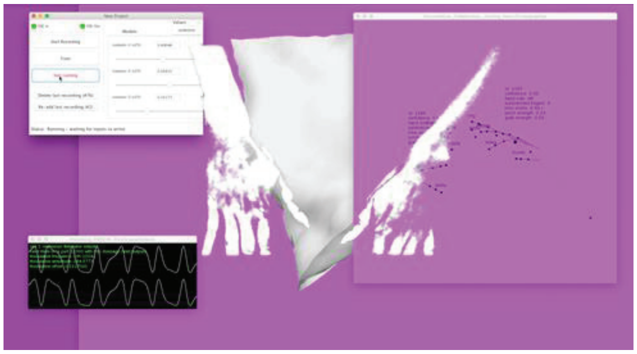
This design approach generated new ideas and conversations within communities typically excluded from the AI Design space (such as individuals working multiple jobs, or those with no tech background). Our goal was to create an accessible project for individuals new to the machine learning space. Rather than optimize the existing neural network, we created a simple example, using Wekinator and collaboratively-sourced materials that showed the basic functionality of machine learning. The community of AI researchers co-developed specific facial movements of protests, inspired by the physical behaviors of protests, from the rhythm of the face movements matching the sounds of a rally to the movement of the eyebrows. The face in the app became the body in the plaza. New movements and protest behaviors emerged based on collaborative thinking in the physical space, along with ways to magnify the impact through machine learning models and collaboratively designed outputs.

One interesting observation that emerged from this research was that individuals liked to engage with models created by other people, often passing a laptop around. Another insight occurred when this project was collaboratively prototyped: new interactions and movements continued to occur as the participants observed each other and became more playful with their creation of training data. Also, in the design process participants wanted to design for multiple modes of presence (X Reality), in both the virtual reality, augmented reality, web and physical experiences. The possibility of porting one behavior across multiple representations of presence could result in interesting design opportunities, within alternative spaces or produce new experiences in the physical space.

Intelligent Protest is an example of embodied community-sourced AI Design, where the outcome designs for multiple bodies engage in a shared goal of protest. Our AI Design resulted in rethinking the Knowledge Design of protest.

Accumulative Collaboration

In "Accumulative Collaboration" we chose a specific audience who attended a performance art event as co-creators. The community co-creation was conducted successively with thirty participants contributing hand gestures, one after another. One "station" containing a computer, cameras, Leap Motion controller, and Wekinator was set up during the performative art event, which enabled us to perform as researchers, facilitating conversations about how these systems may apply, and enabled the participants to observe each other contributing movement data sets through improvised hand motions. This format allowed for a different form of conversation and play because the co-creators were able to observe others creating training data. For example, while the Leap Motion itself affords the usage of hands, the hand improvisations became more interesting when participants began designing with other body parts, such as their feet, or when two participants started training the data together—each using one hand. Such unexpected, impromptu moments arose out of the performativity of this research format, which offers a method for creating more personalized algorithm designs by specialized audiences (such as artists, athletes and so forth). In other words, this research format allowed us to explore what it means to create group-specific or individually curated algorithms by specific nontraditional users.



Research for "Accumulative Collaboration." Community artists engaged with Wekinator to create training examples.

In "Accumulative Collaboration," we collaborated with an open-source machine learning tool Wekinator (Fiebrink 2009) to facilitate human-to-human connections, human-to-machine interactions, and the creation of embodied training data. The research was conducted in a domestic space as part of a curated performance art event. Participants performed improvised hand gestures with the goal of training the open-source machine learning neural network in succession. Thirty participants contributed three hand improvisations each. Each participant's improvised contribution built off the next, creating a growing chain of gestural data and a neural net, thus an accumulation of collaboration. The community creation focused on designing with artists only, a unique collaboration in that it did not focus on one final output, but rather produced conversations and approaches to training data outside of the intended design of the inputs.

To create this accumulative collaboration, we set up Wekinator to receive 15 input values (using the LeapMotion_Fingertips_15Inputs Processing program) and computed 3 continuous output values which were mapped to sound outputs using the Processing_FMSynth_3 ContinuousOutputs mac executable. We selected the neural network algorithm option in Wekinator and defined the ranges for the sound output. Using Leap Motion, participants improvised gestures with their hands to provide unique movement inputs. Hand improvisations became training data for the model, and a duet between machine and human ensued. This approach facilitated an accumulative choreography—one participant followed another, building off previously improvised hand gestures. The ensuing contagion of choreography brought participants (strangers to one another) into a collaborative relationship facilitated by AI.

"Accumulative Collaboration" asks what it means to conduct co-creation of and/or testing of intelligent systems through an accumulative approach. Using this embodied approach to conduct machine learning training results in new playful opportunities with the data, and new design opportunities emerging from training with different kinds of bodies. Thus, the bodies of a given community can be utilized to prototype machine learning systems that can more easily address outliers and design challenges, rather than simply designing with analytic data with which the community has little physical connection to. The benefits of this approach is to engage in useful, inclusive, community-specific AI Design.

Conclusion

Under the umbrella of a concept we call Knowledge Design, we have demonstrated an approach to AI Design that addresses culture, civic engagement, and human-to-human and human-to-machine interactions. We argue for an embodied collaborative knowledge to inform how we engage in AI Design. We present our experimental prototypes and co-creative and embodied research methods to share our vision of an AI Design practice based on Knowledge Design. We used an embodied approach to conduct machine learning training because the data it generates is community sourced. Different bodies, skin tones, and types of faces can be challenging when designing facial recognition systems utilizing computer vision. Using an embodied approach allows AI Designers to design with different bodies. Keeping data diverse from the onset makes it easier to design for those opportunities as they arise.

In the project "Accumulative Collaboration," we explored what it means to engage in collaboratively trained (curated) algorithms and design. In "Intelligent Protest," we engaged in Knowledge Design to create an AI Design project to prototype a new way to protest across spaces. Our prototyping has focused on neural networks. From a technical perspective, we would like to continue to prototype with our community on "Accumulative Collaboration" and "Intelligent Protest" in Wekinator by modifying the neural network algorithm and refining hidden layers, nodes, and training data to create an optimal model for collaboratively preferred output. Additionally, we intend to collaboratively design with linear and polynomial regression algorithms to probe new design opportunities. Overall, the focus of our work is not only to make AI Design more accessible to individuals distanced from AI, but also to create inclusive intelligent products and thinking in the Knowledge Design space.

Acknowledgements

We thank the following for their support and collaboration: our ArtCenter College of Design advisors Anne Burdick, Phil Van Allen, and Elise Co; Kimberly Bridgewater, Morgan Fogarty, Samantha Goodman, Darby Kelley, Lindsey Lollie, Darrian O'Reilly, Crystal Sepúlveda, Alex Shilling, Jay Hong, editors Judy DeTar and Pamela Grieman, Rebecca Bruno of HomeLA, Feminist AI Projects: Bits & Bytes AI Design Research Group, Women Who Code, Rails Girls, The Women's Center for Creative Work, Cross Campus DTLA, FemBots, FemTechNet, The Altadena Public Library, Machine Project, WIML, voidLab, and ArtCenter College of Design Media Design Practices. For their inspiring work, we thank N. Katherine Hayles, Dara Blumenthal, and Sara Ahmed.

References

Adam, A. 1998. *Artificial Knowing: Gender and the Thinking Machine*, 86. London, U.K.: Routledge.

Blumenthal, D. 2014. *Little Vast Rooms of Undoing: Exploring Identity and Embodiment through Public Toilet Spaces.* London, U.K.: Rowman & Littlefield, 47.

Byrnes, N. 2016. Why We Should Expect Algorithms to Be Biased. *MIT Technology Review*.

Dourish, P. 2001. *Where the Action Is: The Foundations of Embodied Interaction.* Cambridge, Mass.: MIT Press, 20.

Fiebrink, R. 2009. Wekinator [Computer software]. Retrieved from http://www.wekinator.org.

Fry, B., and Casey, R. 2001. Processing [computer software], version 3.3.6, December 2017. Retrieved from http://www.processing.org.

Kogan, G. 2015. ofxFaceTracker [Computer software]. Retrieved from https://github.com/genekogan/ofxFaceTracker.

Lewis, C. and Rieman, J. 1993. *Task Centered User Interface Design: A Practical Introduction*. Boulder, Colo.: University of Colorado Department of Computer Science.

Leap Motion [Computer software and controller]. San Francisco, Calif.: Leap Motion, Inc.

Meinders, C. 2017. Embodying Artificial Knowing. M.F.A. diss., Graduate Media Design Practices, ArtCenter College of Design, Pasadena, CA.

Nilsson, N. 2010. *The Quest for Artificial Intelligence: A History of Ideas and Achievements*. Cambridge, U.K.: Cambridge University Press, xiii.

openFrameworks Community. 2004. openFrameworks [Computer software], MIT License. Retrieved from http://openframeworks.cc.

Phillips, P. J., Jiang, F., Narvekar, A., Ayyad, J., and O'Toole, J. 2011. An Other-Race Effect for Face Recognition Algorithms. *ACM Transactions on Applied Perception* 8(2):18–19.

Ratto, M. and Boler, M. eds. 2014. *DIY Citizenship: Critical Making and Social Media*. Cambridge, Mass.: MIT Press.

Sweidan, S. 2016. Improvisation, Quantum Data, Wandering. M.F.A. diss., Graduate Media Design Practices, ArtCenter College of Design, Pasadena, CA.

Wang, Y., and Kosinski, M. 2017. Deep Neural Networks Are More Accurate Than Humans at Detecting Sexual Orientation from Facial Images. Retrieved from psyarxiv.com/hv28a.

Challenges and Methods in Design of Domain-Specific Voice Assistants

Sarah Mennicken, Ruth Brillman, Jennifer Thom, Henriette Cramer
Spotify
{sarahm, brillman, jennthom, henriette}@spotify.com

Abstract

Most of the currently existing voice assistants, like Alexa, Siri, Google Assistant, and Cortana, are generalists. They act as a unifying voice interface to a myriad of controls but rarely support domain-specific expert functionalities. There are efforts to provide more targeted assistant experiences and capabilities around specific areas of applications. In this paper, we discuss several challenges and opportunities in the design of domain-specific voice assistants. We outline a variety of methods to create and utilize an understanding of domain-specific user language and ideas to prototype and study the envisioned user experiences.

Introduction

Amazon's Alexa, Apple's Siri, Google's Assistant and Google Home and Cortana are well-known examples of general-purpose assistants created with the expertise and data available to major tech companies. In this paper, we give a high-level overview of a variety of design challenges, and make the distinction between designing for a general-purpose assistant as opposed to a domain-specific one. By domain, we mean the types of expertise handled by the assistant. A general-purpose assistant, such as Alexa, Google Assistant, and Siri, works across domains such as providing the user with weather information, setting timers and reminders, driving directions and shopping. General-purpose assistants, by necessity, must cover a broad and wide territory of expertise. A domain-specific assistant is a specialist in one particular area, such as a customer service agent like Nuance's Nina (Nuance Press Release 2017) on Alexa, a banking agent or a music service providing personalized music experiences.

It is worth noting that the challenges discussed in this paper apply to many domain-specific assistants, regardless of the machine learning models that power them. Consider an automatic speech recognition (ASR) component: the ASR will likely have to be optimized to correctly transcribe important, domain-specific words, accounting for differences in accents and possible mispronunciations. However, this challenge will present itself regardless of which machine learning techniques are used. While the specific way to implement a solution to a given challenge may depend on underlying techniques and modeling decisions, the occurrence of the challenge should not be.

A challenge for general-purpose voice assistants is that they need a wide breadth of data. This can include audio data, transcribed text, annotated and labeled text for natural language understanding and knowledge graph inputs. Domain-specific assistants, however, come with different expectations, and require a narrower and simultaneously deeper dataset for training and testing. In this paper, we discuss considerations on how user data can be leveraged to identify what aspects to consider for data collection and how to drive prototyping efforts for the efficient transfer of insights to models and technology.

What makes a voice assistant?

From a technical perspective, a voice assistant is a natural language processing pipeline. It consists of many parts, including automated speech recognition (ASR), natural language understanding (NLU), natural language generation (NLG), and text-to-speech (TTS). It can include search, knowledge graph and agent back-ends, as well as agents of different platforms, all of which have to interface with different natural language components. From a user perspective, design has to consider the expectations the user has around how s/he can phrase her questions to the assistant, the functionality it offers, and how it sounds when it responds. This includes the words the assistant chooses and the sound of its voice. The design and functionality choices will affect how users continue to interact with it, similar to how the voice and the vocabulary of another human affect how someone interacts with them. E.g., if one hears the voice of a child using the vocabulary of a 5-year-old, they will adjust their expectations and their own language. People always attribute personality traits to speech, even if it is synthesized by a computer (Nass and Lee 2001). Therefore, it is an important first step for design - even before leveraging user data - to define the role the assistant should convey and a few, core personality traits. Whether the role is to be a representative of a brand or an individual with their own opinions and values will affect how interactions need to be designed. Identifying the target user helps create a user-centered plan for design. E.g., a music companion for teenagers will require a different approach than an assistant for medical support for the elderly.

Domain-specific behavior and expectations

With the role and the target user group in mind, we can create a better understanding of the domain. This helps to anticipate the voice input the assistant will receive and provides insights on expectations. This includes not just expectations for functionality, but also for the tone of voice and the behavior of the assistant. What questions help to guide this process and what data can be leveraged?

How do people talk in the domain that needs to be modeled for a domain-specific assistant?

Examining existing data from **other systems in the same domain** is a useful, though often times not comprehensive, method of understanding what kinds of voice requests the system is likely to receive. For example, text search interfaces often compel users to search for named entities. However, voice requests can often be nonspecific or generic, such as asking a TV assistant “Play a dark and gritty documentary” or “Show me something my friends will like.” Understanding the way that people talk about the target domain is a necessary first step to predict and prepare for the types of voice-specific utterances the system will need to be able to process. Similarly, back and forth dialogues that are crucial in domains such as customer service, cannot per se be derived from non-dialogue, search-type data.

Voice assistants often take on roles that are inspired by existing human roles or tasks, including trying to replicate their domain-specific knowledge, e.g. a travel, or customer service, agent. Both **content analysis**, as well as qualitative design research methods, like **interviewing domain experts** can provide a more comprehensive picture of what utterances to expect or what functionalities to include. E.g., asking experts which questions they are asked by their audience or which questions they would like to ask from individual users, but cannot scale. Dialogues are crucial to understand in fields like customer service, in which case in-depth content analysis of existing interactions is also vital. Creating this understanding of the role the human assistant takes on can help to identify interaction flows. This approach has also been successfully applied for years in the context of information retrieval, e.g., to identify information seeking behavior at a library (Taylor 1968). **Crowdsourcing** can be used in multiple ways, and is a common element of voice projects. First, it is a useful method to elicit large amounts of data to bootstrap natural language understanding systems (Callison-Burch and Dredze 2010). Data can be collected from a variety of crowdworkers from various geographies and domain-related skillsets to increase diversity of training data. Second, it can be used to collect speech data from a diverse population so that a broadly applicable ASR system can be trained and developed (Pavlick et al. 2014). Finally, crowdworkers can label data for supervised machine learning methods and therefore improve existing models. A better understanding of the domain will also help to put user utterances into context. E.g., certain user utterances that seem offensive might be sincere requests in the context of music and entertainment. Content like the song “F*** you” by CeeLo Green or the TV series “I love D***” illustrates this potential ambiguity well. Culturally specific references carry the potential for this ambiguity, too. If new entities are regularly added to content catalogues where popularity fluctuations are frequent, this becomes even more challenging.

What behavior do people expect?

People might have built up expectations from experiences with people in the roles that the assistant is intended to take on, including what the assistant should be capable of, the tone of its interactions, its demeanor, or even how it looks. E.g., consider the stereotypical differences in how people think a travel agent, a bank teller, or a DJ might behave or appear. Of course, the previously mentioned interviews with domain experts provide insights into this, too.

Another way to elicit understanding via qualitative design research is to ask participants to **role play**. Role-playing through Wizard-of-Oz set-ups can identify whether a scripted dialogue works and a more open-ended setup can help identify potential functional challenges. Pretend-users might come up with requests that the Wizard-of-Oz prototype may not be able to solve. Take for the hypothetical example of a restaurant recommendation assistant. The pretend-user might want to “Send that restaurant to my friend Frank”. This could lead to the realization there is no script for sending recommendations to unknown friends. Maybe, the pretend-user would want to “Order me a pizza”. Potentially, being outside the originally envisioned functionality, that might point to a missed opportunity and/or required features, like having to have credit card information on file, and the need to integrate a secure payment partner.

Identifying domain-specific challenges

While people might only expect domain-specific services from a human expert assistant, we cannot necessarily make that assumption for domain-specific voice assistants. One might expect such an assistant to be able to navigate, find music, or even order pizza. Therefore, creating a good understanding what the user expects within a specific domain might provide an initial assumption on what variety of utterances to expect and then to define how you want to deal with the functional limitations of your system.

Functions and knowledge

Deciding how to limit the scope of the NLU/NLG system is a particular problem for domain-specific assistants. For example, while playful questions such as "Are you married?" happen in general assistant contexts, they are less expected for most domain-specific voice assistants. Domain-specific assistants require design and engineering decisions about how and where to limit conversation and how to distinguish erroneous and out of range requests, both of which are potentially unsupported by the machine learning model underlying the assistant. There are several options. The assistant can respond in a way that shapes expectations moving forward (“Sorry, I can’t help you with that.”) at the risk of being perceived as incomplete or less competent. The assistant can use an unsupported utterance as an opportunity to educate the user about what it can do (“No can do. But I can sure be of assistance if you want to book a flight.”).

The design decisions above also open up broader questions about the nature of conversations. For example, if the semantic processing component of an NLP system depends on a knowledge graph, it needs to be decided how to limit its scope. Similar questions arise when building out dialogue management systems, regarding what facets of conversation the assistant should and should not support, such as multiple turn question answering sessions.

Pronunciation

Since users might expect domain-specific assistants to have deep expertise in that domain, unique and little-known terms that do not occur that frequently in general natural language corpora will need to be modeled. Unique terms and pronunciations are not always easily covered by off-the-shelf lexicons. Some domains include entities for which full names are not originally intended to be pronounced, such as emoji in text and email messages. For instance, music systems will have to handle a diverse catalog with unique artist and track names. User-generated music playlists can have names that consist of emojis (Spotify Blog 2017) or make use of character substitutions ($ for S) that might not correspond to obvious pronunciations. Non-obvious and ambiguous pronunciations pose a challenge for ASR systems, and their detection may require dedicated new techniques.

In the music domain, code-switching between languages occurs when users ask to listen to music in multiple languages in addition to their primary language (e.g. “Play *Me gustas tu*”). This poses another challenge for ASR. In the travel domain, an assistant that supports international travel will likely have to train its ASR on more than one pronunciation for international destinations (e.g., the English and Spanish pronunciation of cities in Latin America), and understand that multiple names, across multiple languages, refer to the same location.

Privacy

Hands-free voice assistants also face particular design challenges surrounding confidential information, especially if the assistant is developed for a domain where privacy is highly prioritized, like banking or healthcare. Password controls are challenges for all assistants, but financial assistants may face greater challenges surrounding information such as bank account numbers and sensitive social information such as account balances.

Default behavior

Different assistants also trigger assumptions of a default action on the part of the user. While many utterances contain a verb, some utterances are simply the name of an entity the user would like to search for, similar to text searches. For example, instead of saying “Play David Bowie,” they may just say “David Bowie.” If a user is interacting with a music assistant, these utterances should probably result in a David Bowie album being played. However, on a movie assistant, this utterance may result in the user watching the movie *Labyrinth.* This is different from what a general voice assistant would return; all current general voice assistants that have been brought to market support a general search as their default action. The fact that a user might reasonably expect identical utterances to result in distinct content across different types of assistants poses challenges for assistant design and user research. This also influences the type of linguistic utterance data the model should be trained on and expect to receive.

Prototyping tangible experiences

Depending on the domain and the domain-specific challenges there are easy ways to prototype early on to inform further iterations and refinements.

A quick way to test an envisioned interaction is working with writers who are experienced in writing dialogue and then ask participants to provide feedback. However, this method put participants into a passive position where they act more as an observer, rather than being immersed themselves. **Scripted and pre-recorded dialogues** can alleviate this to some extent. Asking participants to read out the requests or questions that have been identified as common for the domain and then present them with pre-recorded audio will create at least some level of immersion.

Off-the-shelf conversational prototyping tools, such as Alexa Skills or Google Actions, are simple software toolkits for the commercially available hardware Amazon Echo and Google Home. They provide a lightweight way to prototype a dialogue experience for a wide general audience. The main benefit of this type of prototype is a relatively easy setup. However, these platforms are not fully customizable and will not allow designers to model the depth needed for realistic interactions with a domain-specific assistant. They also do not allow full access to the user utterances and speech data that is collected by the hardware which might be required for the prototyping of the envisioned functionalities.

Custom-prototyping tools such as **Wizard-of-Oz tools** for rapid prototyping and testing are widely known in research, but quick-and-easy tools are not yet easily accessible to industry product teams. Oftentimes, a lot of custom work is required to implement such prototypes. Active research is, for example, ongoing in developing in-car voice interfaces (Martelaro and Ju 2017). The recent attention to the 'fake autonomous car' (Solon 2017) in which prototyping involved someone dressing up as a car seat inspired by the Stanford Ghost Rider set-up (Rothenbücher et al. 2016) is a testament to the offbeat creativity still necessary in testing people's reactions to new applications.

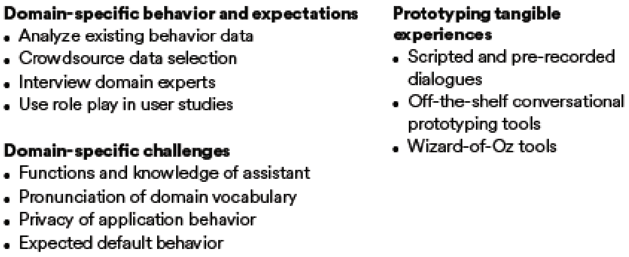
Integrating Machine Learning into Design Prototyping Tools

Making the compelling collaboration between careful interface design and the capabilities of machine learning tangible for user testing is quite challenging. Rapid iteration and prototyping of experiences is a vital process of the design process for voice assistants, but design prototyping tools often do not include the functionalities enabled by machine learning, e.g., personalized or context-aware context. When users are instead presented with data prepared ahead of time or if the interaction lacks personalization, it reduces how representative these studies are for the envisioned experience. The value of experiencing a prototype which includes the models being worked on is therefore significant for end-user testing and to inform iterative design.

Integrated prototyping also allows for a better understanding how design decision result in technical implications. If machine learning-based functionality is a part of the design prototypes, it provides an opportunity to learn about potential errors and edge cases early. User studies with such integrated tools can provide insights into possible limitations that might occur when being used in a real or slightly different context than the one for which the models have been trained for.

In Short

When designing for domain-specific voice assistants, there are many ways to learn from how users and people in roles similar to the assistant naturally speak within that domain and what their expectations are.



Domain-specific behavior and expectations

- Analyze existing behavior data
- Crowdsource data selection
- Interview domain experts
- Use role play in user studies

Domain-specific challenges

- Functions and knowledge of assistant
- Pronunciation of domain vocabulary
- Privacy of application behavior
- Expected default behavior

Prototyping tangible experiences

- Scripted and pre-recorded dialogues
- Off-the-shelf conversational prototyping tools
- Wizard-of-Oz tools

This will help to identify unique challenges that affect what the assistants should be capable of early on and allow for informed design decisions to deal with functional limitations. By including the functionalities enabled by machine learning in prototypes early on will allow collecting more representative user data while also informing and testing machine learning models.

Biographies

Sarah Mennicken

I am a research scientist as Spotify focusing on the design of voice output and novel voice experiences. I work at the intersection of user research, design, and technology helping to translate insights and designs into prototypes that allow studying tangible experiences.

Prior, I was a senior UX scientist/strategist at a startup and a visiting researcher at Microsoft Research focusing on interactive technologies based on real-time computer vision, applied machine learning, and sensing. My academic background and longstanding interest lie in user-centric experiences and interaction design for automated systems and agents, especially in the domestic context.

Ruth Brillman

I'm a research scientist at Spotify focusing on the human side of machine learning with a particular focus on voice systems, natural language processing and the relationships between NLP systems and the linguistic data they rely on for training. I graduated from MIT with a PhD in Linguistics in 2017, and have also done research and development work at Amazon and Akamai.

Jennifer Thom

I am a sr. research scientist at Spotify also focusing on the human side of machine learning and in particular, the social and collaborative aspects of the work conducted by those who label and collect the data that underlie these systems. I'm also interested in conversational interfaces and the social aspects of dialogue between humans and machines.

Previously, I was a research scientist at Amazon where I used various crowdsourcing techniques to provide data to improve the machine learning models that power the Alexa assistant and investigated informal question-answer behavior while a research scientist at IBM Research.

Henriette Cramer

I'm a research lead at Spotify, where I focus on the dialogue between people, data and machines. Prior, I researched user engagement at Yahoo and was a researcher at the Mobile Life Centre in Stockholm where I led projects on human-robot interaction and location-based services. My academic background is in people's responses to adaptive and autonomous systems.

References

Callison-Burch, C. and Dredze, M., 2010, June. Creating speech and language data with Amazon's Mechanical Turk. In Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk (pp. 1-12). Association for Computational Linguistics.

Nuance Press Release., 'Nuance Introduces Nina for Amazon Alexa, First Enterprise Virtual Assistant for the Smart Home', June 1, 2017, https://www.nuance.com/about-us/newsroom/press-releases/nuance-nina-for-amazon-alexa.html

Martelaro, N. and Ju, W., 2017. DJ Bot: Needfinding Machines for Improved Music Recommendations. *AAAI Spring Symposium '17, UX of ML workshop.*

Nass, C. and Lee, K.M., 2001. Does computer-synthesized speech manifest personality? Experimental tests of recognition, similarity-attraction, and consistency-attraction. *Journal of experimental psychology: applied*, *7*(3), p.171.

Pavlick, E., Post, M., Irvine, A., Kachaev, D. and Callison-Burch, C., 2014. The language demographics of amazon mechanical turk. *Transactions of the Association for Computational Linguistics*, *2*, pp.79-92.

Rothenbücher, D., Li, J., Sirkin, D., Mok, B. and Ju, W., 2016, August. Ghost driver: A field study investigating the interaction between pedestrians and driverless vehicles. In *Robot and Human Interactive Communication (RO-MAN), 2016 25th IEEE International Symposium* on (pp. 795-802). IEEE. Vancouver

Spotify Blog, 'Iconic Playlists: What Emoji Say About Music', May 2, 2017, https://insights.spotify.com/us/2017/05/02/spotify-emoji-music/

Solon, Olivia, 'Why did Ford build a 'fake driverless car' using a man dressed as a seat?, September 15, 2017, https://www.theguardian.com/technology/2017/sep/15/self-driving-car-fake-ford-virginia-tech-man-in-seat

Taylor, R. S. (1968). Question-Negotiation and Information Seeking in Libraries. College & Research Libraries, 29(3), 178-194.

Intelligent Devices Retirement Preserve: (un) Natural Wonders

Michael Milano

Art Center College of Design, 1700 Lida St #2319, Pasadena, CA 91103
michaelmilanodesign.com, cargocollective.com/michaelmilanodesign

Abstract

The following is a synopsis of Intelligent Devices Retirement Preserve: (un) Natural Wonders, a media design installation that invites viewers to think about and question what retirement would be like for artificially intelligent devices. By suggesting something as ridiculous as retirement for an artificially intelligent device, the viewer is forced to think about and question the ethical implications of the work life of an artificially intelligent device. The piece questions whether devices should be worked until they can no longer function, whether they should be repaired or upgraded, and whether they should enjoy leisure time. By proposing this scenario, the piece underscores the need to design the experience not just for the user, but also for the device.

Futures of Artificial Intelligence

Artificial intelligence will be implemented more and more into the daily aspects of the labor industry, both in industrial and non-industrial jobs. Industrial jobs are defined as jobs that would normally be described as blue-collar, and non-industrial jobs as ones that would be normally described as white-collar. It is important to describe them as such because as the workforce starts to incorporate artificial intelligence into labor, the definition of "blue-collar" and "white-collar" will change. This change will be much like how labor changed during the industrial revolution. The change might not be as significant, but it will still represent a significant shift in labor roles. Jobs like programming, once seen as white-collar, will more than likely be split, or completely become blue-collar (Thompson 2017). As this shift happens artificial intelligence will obviously advance, becoming more intelligent and capable of greater and broader tasks and jobs. Artificial intelligence will also have the ability to adapt and learn in response to external stimuli or even in response to the person interacting with it. With this advancement the functionality of a job's day to day routines will start to shift, and it's possible that the algorithm will become more of a co-worker embodied within a device, which also could be seen as a co-worker depending on the relationship the worker has with the device in the task being carried out. In this future, especially with industrial jobs, workers that once did manual tasks possibly with the occasional help of devices, would more than likely have to start to program and fix more engineering-based issues for the device on the floor. This is already evident at the company Festo Robots where employees work with cobots, robots that work alongside workers that are using them, to eliminate the need for the worker to make repetitive motions which normally have led to work-related injuries (Festo Robotics 2018) (Hollinger 2016) Instead the workers are being trained, not replaced, to work alongside the cobots, how to instruct them, and how to program. That way if something goes wrong with, or something breaks on their "coworker", they know how to fix it. By doing this the worker doesn't get replaced, and more importantly, workers have said that it makes the job more interesting and enjoyable. This might lead to an emotional bonding for the worker, with the cobot over time.

Importance of the Park

Intelligent Devices Retirement Preserve: (un) Natural Wonders explores what life would be like for artificial intelligence algorithms, and the devices that they embody, if they had a choice to retire and do whatever they wanted. By envisioning and illustrating the outcomes of their retirement, it presents the question to the viewer: do intelligent devices and the algorithms they run off of deserve down time to themselves or should they work until they are no longer serviceable? If the device is decommissioned do we then move the algorithm and its knowledge to a new device? And what happens, or what do we do, when the algorithm is out of date. Is there some form of archive or place for it to live its life, or do we just delete it and let it vanish?

The piece depicts the various stages of the device's lives. Speculating beyond the device solely being a coworker to human, giving them the ability to choose to find new work, continue to work past their expected work quota or end date, and allowing them to do what they want after that date, further develops what a future relationship between artificially intelligent devices and humans would be like. Going an extra step, beyond devices needing humans to exist, helps build a framework for the viewer to accept and question this new work dynamic between humans and artificially intelligent devices, rather than jumping to the common conclusion that robots will replace humans or other common stereotypes. Just as we are starting to see with the shift in job role of workers, who now need to know how to program and engineer their cobots, we will continue to see shifts surpass this initial step of interaction and job role. Which also puts into question how we define employing smart devices. If smart devices have the ability to choose how they continue their lives at a certain point, like to retire, do they have to have a form of income to retire. If we as a society determined that robots do have a right to retire, Bill Gates' idea of having a robot tax might look more like a form of robot social security tax, where the company is taxed both for the use of the robot and for its retirement. (Delaney 2017)

Exploring these questions aims not to anthropomorphize smart devices, but instead get the viewer to think past the norm of how people currently see robots and algorithms, proposing a future where we care for, and even grant limited humanhood or rights to, devices and acknowledge them as co-workers rather than objects. This has already happened in some capacity, though not in a working relationship, with Hansen Robotics robot Sophia being granted citizenship by the Kingdom of Saudi Arabia. (Dom Galeon 2017) Within a working environment, this relationship has already been reported in companies that use cobots, and train their employees to work with and service the devices they work with (Hollinger 2016). This does not mean that the devices have feelings or emotions, but it puts into question the care and ethics that we consider when looking at the quality of life for the device. (Festo Robotics 2018). (Hollinger 2016)

The close working relationship between humans and artificial intelligence also puts into question the responsibility we as designers have to design an experience not only for the user but also for the algorithm. The ethics of training an artificial intelligence network, what it's being trained on, by whom, and for how long should come into question when developing an artificially intelligent network. A great example of this is Microsoft's Twitter bot Tay which was partially trained by people from the internet interacting with it. Sadly the people interacting with Tay were trolling and teaching it inappropriate and hateful things, which resulted in the bot tweeting in support of Hitler and various other inappropriate things. (Kleeman 2016) (Lee 2016) If Microsoft had allowed Tay to continue to learn, it might have shown us what happens when an algorithm is trained on data that is beyond its initial training scope, which in this case would have been the inappropriate input from the internet trolls, and extending on to a much more general knowledge. These factors, design the experience of that algorithm and need to be considered just as much, if not more, than how the user experiences it.

Intelligent Devices Retirement Preserve invites viewers to think past the typical issues regarding sentience, and smart devices taking over jobs, by depicting the retirement of devices with a tongue-in-cheek attitude. An example of this is the depiction of the (un) Natural Wonders that some devices create in their retirement. These far-fetched examples illustrate what a device and its algorithm are capable of and allude to why they might be entitled to a little down time rather than be forced to work from creation to decommission. If the device's lifespan is not cut off after it is done working, and it is left to operate on its own, no longer being maintained, and allowed it to essentially die on its own, what would it do with that time? An example of this already happening, at least to some extent, is unused satellites in higher orbit, NASA will allow them to exist in orbit. (NASA 2015) After NASA stops using them, they don't always stop working which was proven with the satellite ISEE-3, which was still operating seventeen years after NASA lost contact with it. (Campbell-Dollaghan 2014) The intention behind depicting the more plausible to the absurd is to point out to both the general public, but more importantly the scientific community that is developing, engineering, and experimenting with artificial intelligence, that these questions need to be asked, considered, and planned for, designed for, and to start to take these ideas into account when working in the future of artificial intelligence. Sentience most likely will not happen, but we should design humans into the future roles of robots and artificial intelligence. If the ethical treatment of artificial intelligence is not taken into consideration at the outset, the repercussions could become significant or even dangerous issues when creating the future iterations of artificial intelligence.

Artist Biography

Michael Milano is Masters of Fine Art candidate in the Media Design Practices program at Art Center College of Design in Pasadena, CA. His work focuses on how artificial intelligence and robotics will shift labor in the American workforce. More specifically what the relationship between human worker and robot will become, and how we as designers will need to design the user experience for both the user and the algorithm and robot.

References

(n.d.). Retrieved January 12, 2018, from https://spaceplace.nasa.gov/spacecraft-graveyard/en/

Bionic Learning Network. (n.d.). Retrieved January 15, 2018, from http://www.festo.com/group/en/cms/10156.htm

Campbell-Dollaghan, K. (2014, May 22). NASA's Lost Satellite Just Made Its First Contact With Earth in 17 Years. Retrieved January 12, 2018, from https://gizmodo.com/nasa-is-letting-citizens-commandeer-a-long-lost-satelli-1579851540/1583504727

Delaney, K. J. (2017, February 17). The robot that takes your job should pay taxes, says Bill Gates. Retrieved January 12, 2018, from https://qz.com/911968/bill-gates-the-robot-that-takes-your-job-should-pay-taxes/

For the first time ever, a robot was granted citizenship. (2017, October 27). Retrieved January 15, 2018, from https://futurism.com/for-the-first-time-ever-a-robot-was-granted-citizenship/

Hollinger, P. (2016, May 05). Meet the cobots: humans and robots together on the factory floor. Retrieved January 15, 2018, from https://www.ft.com/content/6d5d609e-02e2-11e6-af1d-c47326021344

Kleeman, S. (2016, March 24). Here Are the Microsoft Twitter Bot's Craziest Racist Rants. Retrieved January 15, 2018, from https://gizmodo.com/here-are-the-microsoft-twitter-bot-s-craziest-racist-ra-1766820160

Learning from Tay's introduction. (2016, March 25). Retrieved January 15, 2018, from https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/

Thompson, C. (2017, June 03). The Next Big Blue-Collar Job Is Coding. Retrieved January 15, 2018, from https://www.wired.com/2017/02/programming-is-the-new-blue-collar-job

Talk to Me About Pong: On Using Conversational Interfaces for Mixed-Initiative Game Design

Afshin Mobramaein, Jim Whitehead, Chandranil Chakraborttii
University of California, Santa Cruz
1156 High St
Santa Cruz, CA
mobramaein@soe.ucsc.edu, ejw@soe.ucsc.edu, cchakrab@ucsc.edu

Abstract

Mixed-initiative game design tools combine intelligent agents and human input as collaboration to create novel and interesting content. Traditionally, these systems utilize graphical control-based interfaces. These interfaces can be complex and not reflective of designer intent. Given these issues we propose exploring conversational interfaces for mixed-initiative game design tools. We propose a case-study involving a system for co-creating variations of the game Pong as an initial step towards the exploration of the topic. In addition, we present some of the issues involving the design and implementation of conversational interfaces in mixed-initiative game design tools.

Motivation

From the integration between man and machine envisioned in "Man-Computer Symbiosis" (Licklider 1960) to the sketch-based interactive design capabilities of the soft architecture machine (Negroponte,1975), researchers have long sought tools that create a design collaboration between people and computers. Such systems today tend to be called either *mixed-initiative*, emphasizing the nature of turn-taking between human designers and computer designers, or *co-creative*, emphasizing the contributions of humans and computers without necessarily implying a turn-taking approach.

Mixed initiative design systems have attracted considerable interest within the procedural content generation (PCG) for games community. Traditionally, procedural content generation within games has focused on creating game content---such as a game level or map---with little to no input from the player. The classic dungeon crawler Rogue exemplifies this approach, with each level of a dungeon being generated by a computer algorithm, with no input from the player (Toy et al. 1980). Designer aesthetics are embedded in the generation algorithm. The player either accepts a generated dungeon and plays on, or rejects it by ending their game session. In contrast, a mixed-initiative procedural generation system creates a collaboration between a human designer and a procedural generation algorithm. The Tanagra (Smith,Whitehead, and Mateas 2010) system demonstrates this in the arena of level design for 2D platformer games similar to Super Mario Bros (Miyamoto, Yamauchi, and Tezuka, 1985). The human designer places one or more platforms, thereby creating a partial level design. Tanagra's generator reacts to these platform placements, and automatically generates a suggested design for the remainder of the level. This suggested design can then be modified by the designer, leading to further design suggestions, and so on. Other recent examples include Sentient Sketchbook (Liapis, Yannakakis, and Togelius 2013), a strategy map design tool, Casual Creators (Compton and Mateas 2015), and mixed-initiative game design tools for mobile devices (Nelson et al. 2016,2017).

Current generation mixed-initiative design tools for games provide significantly enhanced design support over traditional tools that provide a blank canvas to human designers. However, there are several ways one might ideally like to improve these systems. First, existing tools constrain and channelize design activity via their user interface affordances. For example, in Tanagra the UI only permits the manipulation of platforms and placement of non-player characters, thus limiting design activity to these facets of gameplay. Second, existing tools don't have a rich model of designer intent, and this limits the kinds of design assistance they can provide. Tanagra's model of designer intent is limited to the platforms placed by the designer, and the notion of "pinning" a platform to a fixed location. Whether the human designer is creating a fast-paced hard level, or a slow-paced easy level is beyond Tanagra's understanding. Finally, lacking a model of intent, it's not possible to manipulate designer intent over time. It isn't possible to ask

Tanagra to make a level "more frantic" or to interpret suggestive but ambiguous desires like "make it colder".

Another challenge with traditional mixed initiative design tools for games is their interface complexity. Designers working with these tools explore a high dimensional design space. An example of this is *Cillr:* A mixed-initiative game creation system (Nelson et al. 2016,2017). This system had an interface with 284 controls, one for each feature of their knowledge representation for games. Nelson et.al. mention the difficulties during user testing relating to difficulty navigating the UI and understanding of the design space that stem from the high dimensionality of the design space, and the complexity of the user interface in the system.

This presents an opportunity to explore different interfaces in the creation of mixed-initiative PCG systems. Conversational interfaces in this case can provide an alternative to GUIs with a large amount of fixed controls presented at once to the human designer. The dialogue-based paradigm of mixed-initiative design is well suited to the turn-based interaction of conversational interfaces. Human designers can take advantage of the conversational nature of these interfaces to explore the design space of an artifact by moving one characteristic at a time in an incremental fashion until they reach their objective. This step-by-step design space exploration combined with a real-time visualization of the generated artefact as it changes throughout the design process can provide an alternative to the complex UIs that mixed-initiative systems use

One use scenario of conversational interfaces in mixed-initiative design system is the one of co-creative game design. Games as a finished artifact are generally described by human designers and users in qualitative terms, rather than quantitative. One can think of describing a video game as "frantic", "smooth", or "stressful" but rarely one describes games in numerical quantities and parameters. As such, using mainly quantitative values while exploring the space of generated games in a mixed-initiative tool might frustrate the human designer during the process. On the other hands, iteratively exploring the design space by describing what aspect of the game is being explored at a time might prove more useful to the human designer. One could think of modifying a parameter of a game by saying "*Make the character move faster"* feel more appropriate as a descriptive characteristic of a game in its design process rather than quantitative descriptions like "*character.xSpeed = 32"*. The former type of interactions in the design process of games lends itself as an opportunity to explore the usage of conversational interfaces in mixed-initiative game design.

Pong as a Reference Problem for Mixed-Initiative Game Design

One problem domain for voice driven mixed initiative design is generating interesting variants of an established game, such as Pong, according to human designer intent. The choice of Pong (Alcorn 1972) as a game domain for mixed initiative design is the one of having a lower design space dimensionality compared to other video games. The space is small enough that interface design complexity issues are not a cause for trouble, but also one large enough to provide interesting variations of the games to human designers.

The design space of Pong can be expressed at both the mechanical level (paddle speed, number of paddles and balls…) and the sub textual levels (what do the paddles and balls represent). For example, the Atari game Video Olympics (Decuir 1977) is comprised of several mechanical variations of the game such as "Super Pong" as well as sub textual variations of the game like "Soccer" and "Handball". A more modern exploration of the design space of Pong is the game "Pongs" (Barr 2012) which provides both types of variations outside of the hardware limitations of previous Pong variant games.

The richness of variations of the game's design space lends itself as an interesting use case for voice driven mixed initiative design. A human designer could try to execute their vision for different kinds of games based on the assumptions provided by the base game. One could imagine a designer collaborating with the system to create a version of Pong that could be described in qualitative terms such as "angry" or "bucolic" by means of a conversational interface to the system. Since mixed-initiative systems employ a dialogue-like use metaphor, the user can explore the design space of the game in a manner that results more "natural" to their design process. Given that the number of agent types, player actions, and physics parameters in Pong is well defined, we can apply a set of descriptive adjectives to the actions that can reflect a human designer's intent during the process. Phrases such as "*I would like to control 5 fast paddles at once"* or "*Make the ball move in a more aggressive manner"* can be mapped to a series of parameter modifications of the game itself in the system. This can lead into a collaborative process between the system and the designer that might result in a more efficient exploration of the design space of the game.

Issues for the Creation of Conversation-Driven Mixed-Initiative Systems

While the usage of conversation-based interfaces might be able to address some of the UI design issues of mixed-initiative game design systems, there are issues to be considered when implementing such a system.

One of the issues of conversational interfaces in mixed-initiative systems is the one of how much can interact with a conversational interface continuously before finding the experience frustrating. This is an analogue to the problem of interface complexity in control-based UIs. While the large amount of controls presented to the user might prove frustrating and hard to navigate to the human designer, an extended interaction with a conversational interface might frustrate the user. This could be interpreted by the designers as the system "not listening" to their input if the results of their conversations about a design do not result in their expected vision of the artifact.

A second issue is the one of finding a starting point between the system and a human designer such that the exploration of the design space of our system leads to a successful co-creation process. This "blank-canvas" process carries several design considerations such as whether either a random solution or a fixed initial point of entry affects the exploration of the design space of our artifacts. In addition, given the conversational nature of the interface the proposition of who initiates the co-creation process arises. Should the designer initiate the exploration of the design space by selecting the parameter they feel is the most appropriate to modify to realize their vision? Or should the system act as a guide by pointing at parameters that might be able to achieve the designers vision in an efficient manner? This is an interesting consideration, since a designer-initiated process might lead to an efficient pruning of the design space of the system, since the user is expected to direct its vision towards the system. On the other hand, a system initiated co-creation process might lead the designer to consider parts of the design space of the system that otherwise would be ignored by letting the system lead the process.

This leads us to the issue of design workflows while using conversational interfaces. One feature that is present in graphical UIs in mixed-initiative systems is the freestyle workflow that having all options presented at once affords the designer. In this sense, a more linear workflow is present in a conversational metaphor. By iterating one aspect of the design at a time in an ordered manner, the human designer might become frustrated by the system. For example, the designer might perceive that they have to methodically go through a phone-tree style menu to reach the aspects they desire to modify. This can become a cumbersome task in the designer's mind as they feel they cannot apply their workflow to a turn-based interaction model. In this sense the system's conversational interface needs to present the affordance of being "freestyle" by letting the designer move around the design space freely in any order.

These above are some of the issues that can arise in the design of conversational interfaces for mixed-initiative co-creative systems. As such, the designer needs to consider these possibilities in order to embrace the advantages that this metaphor affords.

Conclusions and Future Work

We have discussed a proposition for using conversational interfaces in mixed-initiative game design systems. This proposal stems from some of the issues present in traditional graphical UIs used in the design of mixed-initiative systems for PCG. The usage of conversational interfaces that let the user interact with the design space of artifacts, such as games, in an iterative dialogue using qualitative terms presents an alternative to quantitative valued control-based UIs that can address the issues of interface complexity and lack of qualitative manipulation of artifacts present in current mixed-initiative game design systems. In addition, we have presented some of the issues to consider when designing conversational-based interfaces for mixed-initiative design tools such as interaction attrition, starting points, and design workflow issues.

We have started developing a conversational interfaced system that co-creates variants of Pong as an initial exploration of our proposal. We look forward to analyzing the results of user testing of our system with hopes of gaining insights about game design for future systems based on how human designers interact with the system.

References

Alcorn A. 1972. Pong (Game). *Atari 2600*: Atari.

Barr P. 2012. Pongs (Game). *Pippinbarr.com*

Compton K.; Mateas M.; 2015. Casual Creators. In *Proceedings of the Sixth International Conference on Computational Creativity.* Park City, Utah: Association for Computational Creativity

Decuir J. 1977. Video Olympics (Game). *Atari 2600*: Atari.

Liapis A.; Yannakakis G.; Togelius J.; 2013. Sentient Sketchbook: Computer-Aided Game Level Authoring. In *Proceedings of the Eighth International Conference on the Foundations of Digital Games.* Chania, Crete, Greece: Society for the Advancement of the Science of Digital Games

Licklider J.C.R. 1960. Man-Computer Symbiosis. *IRE Transactions on Human Factors in Electronics HFE-1*: 4-11

Miyamoto S.; Yamauchi H.; Tezuka T.; 1985. Super Mario Bros (Game). *Nintendo Entertainment System*: Nintendo.

Negroponte N. 1975. *Soft architecture machines*. Cambridge, Mass.: MIT Press, 1975.

Nelson M.; et al.; 2016. Mixed-Initiative Approaches to On-Device Mobile Game Design. In *Proceedings of the Mixed Initiative Creative Interfaces workshop at CHI 2016.* San Jose, Calif: Association for Computing Machinery

Nelson M.; et al.; 2017. Fluidic Games in Cultural Contexts. In *Proceedings of the Eighth International Conference on Computational Creativity.* Atlanta, Ga: Association for Computational Creativity

Smith G.; Whitehead J.; Mateas M.; 2010. Tanagra: A Mixed-initiative Level Design Tool. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games.* Monterey, Calif: Society for the Advancement of the Science of Digital Games

Toy M.; et al.; 1980. Rogue (Game). *Computer Science Research Group*: UC Berkeley.

Biographies

Afshin Mobramaein is a PhD candidate in Computer Science at the University of California, Santa Cruz. His research interests are in automated game design, assistive AI game design tools, game and media analytics, and procedural content generation.

Jim Whitehead is a Professor in the Computational Media Department, University of California Santa Cruz. He was an active participant in the creation of the Computer Science: Computer Game Design major at the University of California Santa Cruz in 2006. He is the founder and chair of the Society for the Advancement of the Science of Digital Games (SASDG). His research interests include software evolution, software bug prediction, procedural content generation, and augmented design.

Chandranil Chakraborttii is a PhD candidate in Computer Science at the University of California, Santa Cruz. His research interests are in computational models of surprise in games, and game level generation techniques.

How Can I Cook With This: User Experience Challenges for AI in the Home Kitchen

Johnathan Pagnutti
University of California, Santa Cruz
1156 High St
Santa Cruz, California 95064

Abstract

Artificial Intelligence has had an outsized impact on our daily lives, from curating the movies we watch to recommending the books we read. There has been an interest in bringing AI techniques to the kitchen since long before the modern resurgence in AI interest. This is a domain filled with potential victories, with technologies and techniques that are applicable to nearly everyone. In planning a meal, grocery shopping, and even meal preparation, computational systems can assist and empower people to make healthier choices. However, this domain has a unique set of UI and UX challenges that need to be considered that separate it from other applications of artificial intelligence.

This position paper is a proposal for a 20 minute presentation.

Introduction

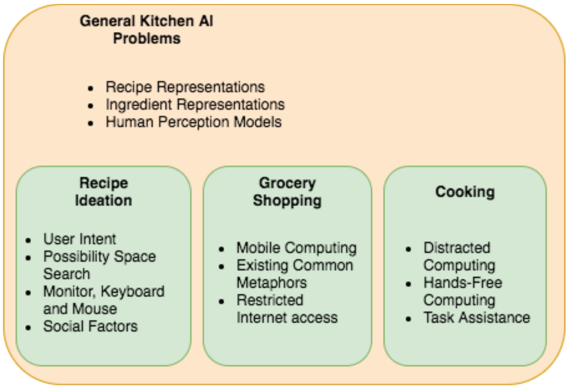
Artificial Intelligence systems have been creating new recipes since CHEF(Hammond 1986), a case-based planner that designed new schezwan recipes. Today, recipe recommendation engines (such as Yummly[1]), databases (such as CocktailDB [2]), and AI platforms (such as Wellio[3]) demonstrate a sustained interest in trying assist and augment cooking tasks.

In addition, HCI has also seen the kitchen as a space for innovating in how we interact with computers. Working with food offers the potential for a design space characterized by *celebratory technology*. Celebratory technology focuses on the positive, successful things humans can do, rather than correcting flaws or mistakes (Grimes and Harper 2008). This concept of technology to celebrate, rather than technology to correct, has also been explored in terms of health (Parker, Harper, and Grinter 2011).

Using AI to promote home cooking and working in the kitchen has high potential health benefits. Eating healthier has been identified as a core component of American wellness by the US Office of Disease Prevention and Health Promotion (ODPHP). Nutrition and good diets are a key component to healthy living(US Department of Health and Human Services and Office of Disease Prevention and Health Promotion 2010). Encouraging people to cook at home has had them feel more in control of their diets and connect with others(Simmons and Chapman 2012).

AI, in a rough sense, is often used to mean 'the automation of intellectual tasks normally performed by humans'. What these tasks are, and the best way to go about performing this automation, is often different from field to field, or even task to task.

I'd like to motivate research by discussing some cross-cutting problems that need to be solved for culinary AI, regardless if the end application is a cooking assistant or a recipe generator. Then, I'd like to focus on three potential tasks a culinary AI may need to solve: meal planning, shopping, and cooking in a kitchen.



Cross-Cutting Problems in Food AI

There are a number of unsolved and open problems that need to be solved at the intersection of AI and food. These problems are linked to all three highlighted sub-domains, and affect even more potential interactions between the culinary realm and AI. Furthermore, this is only a sample of the crosscutting problems, there are likely even more that haven't even begun to be investigated.

Ingredient Representation. How can we represent an ingredient to a computer? Ingredients are more than just plain text list items in a bill for a recipe. They have rich ontological properties (for example, is a Tomato a fruit or a vegetable, and if it is a fruit, does that mean it belongs in a fruit

[1] https://www.yummly.com/

[2] http://www.cocktaildb.com/

[3] https://wellio.getwellio.com/

smoothie?). Ingredients have key sensory properties, such as aroma, taste and mouthfeel that need to be represented to an AI. Electronic noses and tongues, designed to try and replicate how the sensors in our own noses, mouths and tongues work(Deisingh, Stone, and Thompson 2004) have been used in food analysis, but they need to mixed with other sensory data in a unified perception model, as even the way the crunch of food sounds to our ears can change our sensory perception (Deisingh, Stone, and Thompson 2004).

Recipe Representations. A recipe can be thought of as a plan, and there is a long history of planning research. Or perhaps, a recipe is a set of rules, which also has a rich tradition of research in AI. However, we need plans to be adaptable, able to be modified on the fly when you've realized you've forgotten a key ingredient in the store. We also need to adapt these existing bodies of research to culinary constraints, such a difficulty to prepare, preparation time and if part of the plan or rules can be done the night before.

Perception Models. Unlike graphics or acoustics, we don't have an easy numerical representation of aromas or tastes. We need to build models of how humans perceive these things, and how these perceptions change around various contexts and under various chemical interactions. Currently, flavor scientists and product researchers use flavor wheels (Di Donfrancesco, Gutierrez Guzman, and Chambers 2014; Noble et al. 1984) to capture aroma and taste perception. These abstractions describe an ontology of flavor words and relate various terms in space to aid in describing a sensation. These, along with professional and home chefs discussing their trade, can give us a starting, high level abstraction, to work with sensation of taste and flavor.

Food Availability / On Hand Ingredients. Keeping track of the state of a user's pantry is non-trivial. Not all ingredients at the point of purchase will end up in the buyer's pantry, not everything bought at grocery stories is for meals, and food is not stored in a single location. You can arm a fridge with sensors, but what about kitchen cabinets, pantries and counter spaces?

Recipe Planning

Planning what to eat tonight is a significant task, as a meal planner needs to balance personal desires (what do I even want to eat?), on-hand food items (what's in my fridge?), and other concerns (what's the healthy thing to have?). From a UI/UX perspective, it's not unreasonable to expect meal planning to happen in an environment similar to where a laptop might be used. It's a significant task, and users are likely to devote their full attention to figuring out what to eat tonight.

However, meal planning is often not done alone—some discussion with family members, friends or roommates is key in figuring out what to eat tonight. Integrating these communication channels is key to an AI designed to assist with meal planning. Users also need easy ways to search through a vast possibility space of meals, slicing away parts with various constraints. In this, a meal planning AI can almost be thought of as a casual creator (Compton and Mateas 2015), as average users don't need sophisticated design or planning tools to come up with a meal. Casual Creators are generative AI tools that support creativity intrinsically, rather than as an extrinsic way to solve a task. They just need simple interfaces that make exploration and experimentation easy, with rapid feedback and an easy way to share what they've made. These traits are important to the food domain, to help the meal planning activity feel intrinsically rewarding.

It is important to note, however, that Casual Creators are desirable to help people find novel or surprising artifacts in possibility spaces. It has been shown that novelty has an inverse impact on food choice, people actually desire familiarity in their food selections (Meiselman 1996).

Grocery Shopping

AI can also assist in shopping tasks, and the logistical complexity around going from a plan (the recipe we'd like to make) to a collection of ingredients, ready to be prepared in a kitchen. The interface challenges here are somewhat different than the free-form, focused space exploration of meal planning.

While shopping, users are likely to only have a mobile phone on them. Furthermore, grocery stores are not hotbeds of Internet connectivity, and a shopper may not have a connection to a remote server to offload processing tasks or access a database. Users may be shopping for single meals, or maybe getting a large amount of groceries for multiple meals. Although there is a rich history of metaphors for grocery shopping (e.x: shopping lists), do those same metaphors make the most sense when developing an AI to assist with shopping?

Meal Preparation

Can an AI help someone actually prepare a meal? Although there is a push for robotics in the kitchen[4], we'd like to think of how a digital assistant can interact with a home cook to successfully prepare a meal. This environment is hands free, as a cook has their hands full with cooking utensils, ingredients and tasting a bit of what they're preparing. Furthermore, this is a distracted environment, as a cook's attention is focused on successfully making a meal.

Perhaps in the future, a kitchen AI will use integrated sensors in various appliances (an 'embedded' or 'smart' kitchen) to interact with a cook, Conversational AI interfaces seem like a potential huge win here. Some of the current interaction paradigms are very limiting. It's a not uncommon paradigm in current conversational interfaces to stop listening for user input after a few seconds[5]. This is a smart security concern, as people are uncomfortable with an 'always listening' device in their homes. However, for cooking, it's not uncommon for the next request to happen well after the interaction window is up (because a user was doing something else for 15 seconds), which often means a user needs

[4] With companies like Moley Robotics: http://www.moley.com/

[5] At time of writing, the Amazon Alexa conversational interface listed that the next input needed to be within an 8 second window https://developer.amazon.com/docs/custom-skills/custom-interaction-model-reference.html

re-prompt the conversational interface to have it 'remember' where they last were.

Balancing these concerns is paramount to finding an interaction paradigm that works well in the kitchen. We need to think about UX designs that will help people trust computational assistants to bring a meal to a successful conclusion without feeling like backseat drivers.

Conclusions

Culinary AI is fertile ground for new problems in AI interaction. From the many contexts that users may interact with such a system to the interesting constraints within each of those contexts, if we want to bring artificial intelligence to the kitchen, these problems need solutions.

There are very large victories if we can build these sorts of AI assistants. From being able to promote healthier meals, to giving people a sense of empowerment and control over their food intake, culinary AI has the potential to make both physical and mental wellness improvements in a home chef's life.

Author Biography

Johnathan Pagnutti is a Ph.D candidate in Computer Science at the University of California, Santa Cruz with the Augmented Design Lab. His research primarily focuses on developing algorithms that can create new recipes, with the aim of helping more people cook in the kitchen.

References

Compton, K., and Mateas, M. 2015. Casual creators. In *ICCC*, 228–235.

Deisingh, A. K.; Stone, D. C.; and Thompson, M. 2004. Applications of electronic noses and tongues in food analysis. *International journal of food science & technology* 39(6):587–604.

Di Donfrancesco, B.; Gutierrez Guzman, N.; and Chambers, E. 2014. Comparison of results from cupping and descriptive sensory analysis of colombian brewed coffee. *Journal of sensory studies* 29(4):301–311.

Grimes, A., and Harper, R. 2008. Celebratory technology: new directions for food research in hci. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 467–476. ACM.

Hammond, K. J. 1986. Chef: A model of case-based planning. In *AAAI*, 267–271.

Meiselman, H. L. 1996. The contextual basis for food acceptance, food choice and food intake: the food, the situation and the individual. In *Food choice, acceptance and consumption*. Springer. 239–263.

Noble, A. C.; Arnold, R.; Masuda, B. M.; Pecore, S.; Schmidt, J.; and Stern, P. 1984. Progress towards a standardized system of wine aroma terminology. *American Journal of Enology and Viticulture* 35(2):107–109.

Parker, A. G.; Harper, R.; and Grinter, R. E. 2011. Celebratory health technology.

Simmons, D., and Chapman, G. E. 2012. The significance of home cooking within families. *British Food Journal* 114(8):1184–1195.

US Department of Health and Human Services and Office of Disease Prevention and Health Promotion. 2010. Healthy people 2020.

Procedure Automation: Sharing Work with Users

Debra Schreckenghost, Scott Bell, David Kortenkamp, James Kramer
TRACLabs, 16969 N. Texas Ave, Suite 300, Webster, TX 77598
schreck@traclabs.com

Abstract

An area of interest for NASA is the use of procedures as the basis of task automation. The PRIDE software was developed to author and execute electronic procedures for NASA spacecraft and habitat operations. We describe our approach for modeling human-automation work based on a procedure language, and allocating and execution tasks among a human-automation team. We illustrate our approach with examples of collaborative work using procedure automation.

Procedure Automation for NASA

The PRIDE software was developed to author and execute electronic procedures for NASA spacecraft and habitat operations. The nature of work in NASA operations requires specialized knowledge about complex systems that may be used infrequently. Additionally, error consequences when performing the job can be significant. NASA uses procedures as a means of "planning ahead" how operators will perform both nominal and off-nominal work, to mitigate the risks of operating in such a high criticality domain.

Procedures are used to manage spacecraft and habitat systems, perform Extra Vehicular Activities (EVAs), and conduct space science and exploration. Astronauts and flight controllers are trained using procedures. Qualifying for flight control positions includes performance using procedures. Thus, NASA users are familiar with procedures and procedures are well-maintained.

NASA is interested is the use of procedures as the basis of task automation. As astronauts move deeper into space, their workload is expected to increase because Earth-based flight controllers will not be in continuous real-time communication. Task automation has potential to reduce astronaut workload for such missions. It also can improve response time as communication latency with Earth increases. And automation can prove beneficial in performing tasks prone to human error, such as vigilance monitoring.

One challenge in automating procedures is *capturing procedure knowledge* that can be used both for manual and automated execution. These task models often are built when the manual procedure is first documented, and well before the automation is available. Thus, our approach must produce electronic procedures for either manual or automated execution. NASA procedure authors are subject matter experts, so we also need an approach to task modeling that does not require computer programming skills.

Another challenge in automating procedures is *communicating automation behavior* and its effects on spacecraft and habitat systems. The introduction of task automation into NASA operations requires establishing operator trust that automation is reliable and predictable. Even when operating at a high level of automation, it is expected that operators must maintain awareness of automation actions, because they are responsible to direct and manage automation. It also is expected that operators will intervene when automation or system behavior is different than expected.

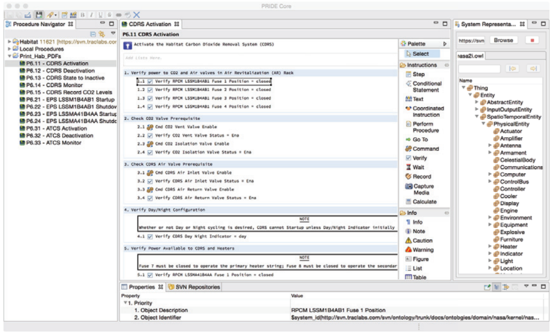
The PRIDE electronic procedure software was developed to address these challenges. It consists of a procedure editor (Pride Author), web-based display server (Pride View), and an automation engine (PAX). We describe our approach for modeling human-automation work based on a procedure language, and allocating and executing tasks among a human-automation team. We illustrate our approach with examples of collaborative work using procedure automation. We summarize our studies of performance with procedure automation. We propose to present our position with demonstration at the workshop.

Modeling Human Work for Automation

Inspection of the procedures used by NASA human space flight reveals an underlying action vocabulary and grammar for using this vocabulary that has a clearly defined semantics. When managing spacecraft or habitats, operators need to perform actions such as 1) send *commands* to a system 2) *verify* sensed values are as expected 3) *record* sensed values at a specific point in the procedure, and 4) *wait for* a sensed value to reach a target value. These atomic actions are composed into checklists with conditional action sequencing, such as 1) performing a subset of actions *conditional* upon situated information, and 2) *looping* through a subset of actions until a condition is true. PRIDE task models are represented using a procedure representation language (PRL; Kortenkamp, et al., 2008) that abstracts this vocabulary in a set of instruction types for building the action sequences seen in procedure checklists.

One user of PRIDE procedures is the procedure author who creates and modifies the PRL. For NASA, procedure authors are subject matter experts. They usually are engineers, scientists, or mathematicians. While they understand how to use a computer, they often have no background in computer programming. They typically use Microsoft Word to author procedure documents that are translated into XML files by a programmer. The Pride Author software provides a way for authors to produce XML directly while manipulating instruction objects (Izygon, et al 2008).

To add an instruction, the author drags the desired instruction type (corresponding to an action) into a central canvas area. This produces an instance of that type. Manipulation of these instruction objects in the canvas automatically produces PRL in the background. What the author sees is an action-object pair similar to what they typed into the Word document e.g., a valve enable command is displayed “Cmd CO2 Vent Valve Enable”. The author also drags items from a model of the system commands and data (called the *System Representation*; Bell, et al., 2015) to insert references to system commands and telemetry verifies. Figure 1 shows an example of the procedure editing user interface for building PRL procedures.



P6.11 CDRS Activation

Activate the Habitat Carbon Dioxide Removal System (CDRS)

Add Lists Here.

1. Verify power to CO2 and Air valves in Air Revitalization (AR) Rack
1.1 ☑ Verify RPCM LSSM1B4AB1 Fuse 1 Position = closed
1.2 ☑ Verify RPCM LSSM1B4AB1 Fuse 2 Position = closed
1.3 ☑ Verify RPCM LSSM1B4AB1 Fuse 3 Position = closed
1.4 ☑ Verify RPCM LSSM1B4AB1 Fuse 4 Position = closed

2. Check CO2 Valve Prerequisite
2.1 Cmd CO2 Vent Valve Enable
2.2 ☑ Verify CO2 Vent Valve Status = Ena
2.3 Cmd CO2 Isolation Valve Enable
2.4 ☑ Verify CO2 Isolation Valve Status = Ena

3. Check CDRS Air Valve Prerequisite
3.1 Cmd CDRS Air Inlet Valve Enable
3.2 ☑ Verify CDRS Air Inlet Valve Status = Ena
3.3 Cmd CDRS Air Return Valve Enable
3.4 ☑ Verify CDRS Air Return Valve Status = Ena

4. Verify Day/Night Configuration
NOTE
Whether or not Day or Night cycling is desired, CDRS cannot Startup unless Day/Night Indicator initially
4.1 ☑ Verify CDRS Day Night Indicator = day

5. Verify Power Available to CDRS and Heaters
NOTE
Fuse 7 must be closed to operate the primary heater string; Fuse 8 must be closed to operate the secondar
5.1 ☑ Verify RPCM LSSMA41B4AA Fuse 1 Position = closed

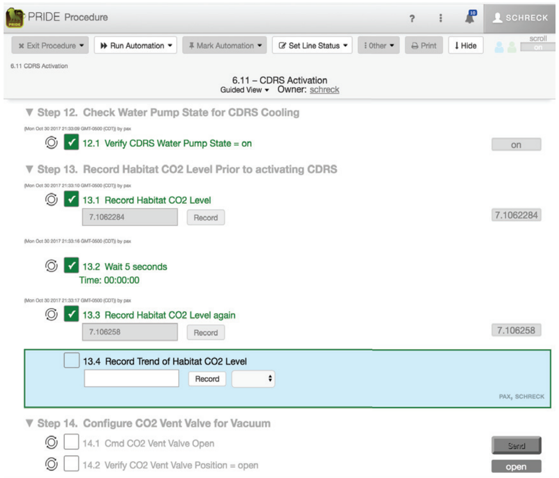
Property	Value
1. Priority	
1. Object Description	RPCM LSSM1B4AB1 Fuse 1 Position
2. Object Identifier	$system_id(http://svn.traclabs.com/svn/ontology/trunk/docs/ontobgies/domain/nasa/kernel/nas...

Figure 1. Pride Author User Interface

When executing the procedure, both the operator and automation use the same PRL task model to perform tasks. This model combines information to instruct a person what actions to take with information needed to execute those actions. Thus, a task to compare a sensed instrument reading to a target value will include both operator directions for what values to compare and data references for accessing current sensed readings. This model is used to generate a web user interface of the procedure document that is directly manipulated by a person to perform the task. The same model is used by the software to automate tasks.

As the procedure instructions are executed, the procedure display is annotated with information about the state of execution (what has been done, what remains to be done); see Figure 2. The same annotations are used whether a person or automation performs the task.

Thus, the same task-based user interface is used to monitor the actions of automation as is used to perform actions manually. This *shared task model* is the basis of human-automation communication about the task. Structuring the work of automation according to human work improves the transparency of automation actions. This approach provides a means for establishing common ground about the ongoing task that should improve operator understanding of automation behavior (Clark and Brennan, 1991).



6.11 – CDRS Activation
Guided View Owner: schreck

▼ Step 12. Check Water Pump State for CDRS Cooling
(Mon Oct 30 2017 21:33:09 GMT-0500 (CDT)) by pax
☑ 12.1 Verify CDRS Water Pump State = on — on

▼ Step 13. Record Habitat CO2 Level Prior to activating CDRS
(Mon Oct 30 2017 21:33:10 GMT-0500 (CDT)) by pax
☑ 13.1 Record Habitat CO2 Level
7.1062284 Record — 7.1062284

(Mon Oct 30 2017 21:33:16 GMT-0500 (CDT)) by pax
☑ 13.2 Wait 5 seconds
Time: 00:00:00

(Mon Oct 30 2017 21:33:17 GMT-0500 (CDT)) by pax
☑ 13.3 Record Habitat CO2 Level again
7.106258 Record — 7.106258

□ 13.4 Record Trend of Habitat CO2 Level
Record
PAX, SCHRECK

▼ Step 14. Configure CO2 Vent Valve for Vacuum
□ 14.1 Cmd CO2 Vent Valve Open — Send
□ 14.2 Verify CO2 Vent Valve Position = open — open

Figure 2. Pride View User Interface

Sharing Task Responsibility with Automation

Shared human-automation work for complex, high risk domains benefits from the ability to tailor the task allocations to the situation. For example, workload balancing may require a redistribution of tasks among the human-automation team. For electronic procedures, this means shifting or sharing the responsibility to perform instructions or make decisions between operators and automation. Each instruction is designated as manual only or automatable. *Manual only* instructions can only be performed by a person. *Automatable* instructions can be performed either by automation or a person. For the domains in which PRIDE procedures have been used, the ability to designate an instruction as *Automated Only* has not been needed. These designations are made when the procedure is authored, and can be adjusted as needed when a procedure is performed (Schreckenghost, et al., 2008).

Responsibility to complete an instruction can be shared by the operator and automation. Instructions have an optional Witness property indicating when a person should approve the action taken by automation before proceeding to the next instruction. Failure of a human witness to approve the instruction is considered anomalous execution.

Procedure instructions are designed to be executed in the order shown in the procedure. When performing instructions manually, however, the user is able to alter the order of execution. PRIDE provides functionality (*oversight mode*) to alert the user when doing an instruction out of order, but such re-ordering is not prevented. When performing instructions automatically, the order of execution is enforced by the automation (*guided mode*). The currently "active" instruction is indicated by a colored, labeled focus bar placed behind the instruction. The operator can only manipulate command buttons or other interaction forms in the active instruction; all other instructions are disabled for manipulation until the focus bar reaches them.

Procedures can be composed of a mix of *Manual Only* and *Automatable* instructions. When operating in guided mode, the automation will pause when it reaches a *Manual Only* instruction. The interaction forms for that instruction are enabled for manipulation. If the user completes the manual action, the focus bar moves to the next instruction and automation resumes, if the instruction is designated *Automatable*. The user also has the option to skip the instruction, fail the instruction, or stop automation.

Examples of Collaboration with Automation

Multiple procedures can execute concurrently, operating at different levels of automation and with different types of human involvement. This supports a variety of human roles when performing collaborative work using procedure automation. We describe some examples of collaborative work with procedure automation below.

Joint human-automation work. Procedure instructions are executed by both the operator and the automation. Tasks are allocated according to policies, such as risk reduction. For example, some NASA operations rely on flight crew to assess the risk of issuing system commands and thus require all commands be sent by a person, while verifies and records can be done automatically. In other operations, human error may pose the greater risk and tasks will be allocated to automation. Allocations may be adjusted differently when executing the same procedure under different circumstance. For example, after changing out a sensor the operator may perform instructions manually that would normally be automated, to ensure that the new sensor behavior matches that expected in the procedure. Figure 2 shows an example of a joint human-automation procedure for starting up a Carbon Dioxide Removal System (Schreckenghost, et al., 2015).

Human supervision of automation. The operator decides which procedures to perform and when to perform them, while the automation executes most of the procedure instructions. Additionally, the human assesses whether automation performance is acceptable. Work design for this style of collaboration includes minimal operator performance of instructions, since the operator's primary responsibility is to manage the work. Often direct intervention by the person is an indication of work breakdown. An example of human supervision of automation is the use of procedure automation to manage the work of an autonomous robot. In one application of the PRIDE software, the operator assigns procedure sequences to a humanoid robot for the purpose of configuring switches.

Distributed human-automation teams. This type of collaboration requires users to perform coordinated work while physically distributed. Procedure automation represents another "team member" available to perform work. For example, all Extra Vehicular Activity (EVA) by NASA astronauts requires two astronauts working outside the vehicle and at least one crew member inside the vehicle or on Earth. For such work, multiple instruction sequences are ongoing concurrently. It is necessary to identify coordination points where these sequences must synchronize. PRIDE can designate instructions as "coordinated," which adds concurrency metadata used during execution. Specifically, it links two instructions in different procedures and identifies whether they should be performed simultaneously or serially. These metadata about coordination points should be respected by both humans and automation.

Performance with Procedure Automation

We have evaluated human performance using PRIDE automation in a number of NASA experiments. To establish a baseline for manual performance we compared manual use of PRIDE procedures with use of an analog for International Space Station (ISS) electronic procedures (Billman, et al., 2014). A key difference between these systems is that live data and commands are embedded in PRIDE procedure displays while data and commands are accessed from a separate display for ISS. Condition effects for both completion time and number of successful users were large enough to be significant for small n (11). Mean completion time was reduced by approximately half. No users had command errors using PRIDE while all users but one had command errors using ISS displays. Next, we compared manual use of PRIDE with PRIDE automation. Preliminary results indicate a reduction in execution timing and workload when using automation (n=27; Holden et al.,

2018) as well as user preference for automation. We also expect performance improvement when users multi-task with procedure automation. We are investigating strategies for work allocation to improve performance when multitasking with automation.

Conclusions and Future Work

PRIDE automation is an example of a knowledge-based system using a hierarchical task language PRL to automate system monitoring and control. It includes rule-based activation of action sequences based on sensed data. Other similar systems include Reactive Action Packages (Firby, 1989), Task Description Language (Simmons, et al, 1998), and Plan Execution Interchange Language (Estlin et al., 2006). Unlike these systems, PRIDE was designed for humans and automation to perform shared procedural work, which requires effective human-automation communication and collaboration. The ability to designate tasks dynamically to either humans or automation is an example of a hybrid human-AI collaboration. Our user interface for procedure automation uses human task models to improve communication of AI behavior to users. All automation actions correspond to actions in human-comprehensible procedures, making these actions transparent and predictable, and potentially improving trust in automation.

While it is possible to reactively select which procedure to automate based on current conditions, PRIDE does not support reactively modifying procedure actions or action sequences. An area for future research is the use of machine learning techniques to adapt existing procedures or create new ones from task observations. Programs such as DARPA's Explainable AI (XAI) can provide techniques for learning procedural sequences that are more understandable and usable by users.

Our development of a procedure editor allowing subject matter experts to author executable procedural task models is an example of a tool for non-AI specialists to build AI models. An area for future research is adding constraint satisfaction tools to help non-AI specialists author procedures that respect domain action sequence constraints.

Finally, the current procedure user interface is intended for monitoring automation while performing low-level actions. For users to multi-task manual procedures with automated procedures, new user interface designs are needed that help users maintain automation awareness without vigilance monitoring of these low-level actions.

Acknowledgements

Human studies with PRIDE were performed with collaborators Dr. Dorrit Billman/ San Jose State University, and Dr. Kritina Holden/ Leidos. PRIDE technology was developed under NASA SBIR. Work on human supervision of automation was supported by the National Space Biomedical Research Institute through NASA NCC 9-58.

References

Bell, S., P. Bonasso, M. Boddy, D. Kortenkamp, and D. Schreckenghost (2015). PRONTOE: An Ontology Editor for Domain Experts. *Communications in Computer and Information Science*, Vol. 454. Fred, A., Dietz, J.L.G., Liu, K., Filipe, J. (Eds.)

Billman, D., D. Schreckenghost, & M. Pardis (2014). Assessment of Alternative Interfaces for Manual Commanding of Spacecraft Systems: Compatibility with Flexible Allocation Policies. *Human Factors and Ergonomics Society Annual Meeting*, Chicago.

Clark, H. H., & Brennan, S. E. (1991). Grounding in communication. In L. B. Resnick, J. M. Levine, & S. D. Teasley (Eds*.), Perspectives on socially shared cognition* (pp. 127--149). Washington, DC, USA: American Psychological Association.

Estlin, T, A. Jonsson, C. Pasareanu, R. Simmons, K. Tso, and V. Verma. Plan Execution Interchange Language (PLEXIL). *NASA Technical Report TM-2006-213483*. April 2006.

Firby J (1989) *Adaptive Execution in Complex Dynamic Domains*, Ph.D. Thesis, Yale Univ. Technical Report #672 Jan 1989.

Holden K. Schreckenghost D. Greene M. Hamblin C. Lancaster J. Morin L (2018). Electronic Procedures for Crewed Missions Beyond Low Earth Orbit (LEO). Presentation at *NASA HRP Investigators Workshop*, Galveston, TX, Jan 2018.

Michel Izygon, David Kortenkamp, and Arthur Molin (2008). A procedure integrated development environment for future spacecraft and habitats. In *Proceedings of the Space Technology and Applications International Forum (STAIF 2008).*

Kortenkamp, D. R. P. Bonasso, D. Schreckenghost, K.M. Dalal, V. Verma, and L. Wang. (2008) A Procedure Representation Language for Human Spaceflight Operations, *9th International Symposium on AI, Robotics & Automation in Space, i-SAIRAS-08*

Schreckenghost, D., R. P. Bonasso, D. Kortenkamp, S. Bell, T. Milam, C. Thronesbery. (2008) Adjustable Autonomy with NASA Procedures. *9th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Pasadena, CA.

Schreckenghost, D., D. Billman, and T. Milam. Effectiveness of Strategies for Partial Automation of Electronic Procedures during NASA HERA Analog Missions (2015*). International Joint Conferences on Artificial Intelligence, Proceedings of AI in Space Workshop* July 2015. Buenos Aires, Argentina.

Simmons R. and Apfelbaum D. A Task Description Language for Robot Control, *Proceedings of Conference on Intelligent Robotics and Systems*, Vancouver Canada, October 1998.

Biography

Debra Schreckenghost is a Senior Scientist at TRACLabs. She conducts research in the areas of adjustable autonomy, human interaction with automation, and real-time performance of robots and automation. Scott Bell is the software lead for the PRIDE procedure system. Jim Kramer developed the PRIDE PAX software. David Kortenkamp is the TRACLabs' President and a key designer of PRIDE.

Assessing and Addressing Algorithmic Bias — But Before We Get There

Aaron Springer,[1,2] Jean Garcia-Gathright,[1] Henriette Cramer[1]
[1]Spotify, 48 Grove St., Somerville, MA, / 988 Market St, San Francisco, CA, USA
[2]University of California Santa Cruz, 1156 High St., Santa Cruz, California, USA
alspring@ucsc.edu, {jean, henriette}@spotify.com

Abstract

Algorithmic and data bias are gaining attention as a pressing issue in popular press—and rightly so. However, beyond these calls to action, standard processes and tools for practitioners do not readily exist to assess and address unfair algorithmic and data biases. The literature is relatively scattered and the needed interdisciplinary approach means that very different communities are working on the topic. We here provide a number of challenges encountered in assessing and addressing algorithmic and data bias in practice. We describe an early approach that attempts to translate the literature into processes for (production) teams wanting to assess both intended data and algorithm characteristics and unintended, unfair biases.

Introduction

There has been around 20 years of early research into the topic of algorithmic fairness and understanding of its outcomes (Friedman & Nissenbaum, 1996). The explosion of widespread machine learning has pushed algorithmic and data bias to the front lines of both the tech press and mainstream media. In parallel, specialized research communities are forming. Promising new initiatives such as the AI Now Institute have been initiated. The FATML workshop has turned into a full conference (FAT*, 2017), a new AAAI/ACM Conference on AI, Ethics, Society has been formed (AAAI, 2017), the ACM has now presented guidelines for algorithmic fairness (Dopplick, 2017). Pragmatically speaking, this increased attention to the topic is great, but these communities' calls to action are still very hard to apply. Pragmatic methods and tools are absolutely necessary to translate nascent research into work in industry practice - also pointed out by Kate Crawford in her WSJ op-ed (Crawford, 2017).

The proliferation of different communities, and the scattered literature presents industry practitioners with challenge to keep up, even when they're highly motivated. Reported studies or methods may also not be fully applicable in practice. We here outline a number of (early) lessons learnt from conversations with Machine Learning-oriented product teams, and thinking through the pragmatic translation of literature into practice.

Background

A wide variety of bias literature and a wide variety of definitions of bias exist. Bias, as a term in Machine Learning contexts, is used in somewhat divergent ways. Bias can be defined as unfair *discrimination*, or it can be framed as a system having certain *characteristics*, some intended and some unintended. Any dataset, and any Machine Learning-based application is 'biased' in the latter interpretation. This means we need to distinguish between unfair/unintended and intended biases. We base our work for practitioners on the pragmatic principle that any dataset is 'biased' in some way, that no dataset completely represents the world, and that human decisions in Machine learning systems inherently have tradeoffs that can result in (un)intended biases. The goal for product teams is to consider which characteristics of data, algorithms, and outcomes are aligned with the goals that they want to achieve—and side—effects.

For the purposes of this discussion, we take specific example definitions and frameworks. We use an adjusted definition from Friedman & Nissenbaum on 'Computational bias' as placeholder for (unfair) algorithmic bias: 'Discrimination that is systemic and unfair in favoring certain individuals or groups over others in a computer system' (Friedman & Nissenbaum, 1996). Where we use 'systemic' in our definition, Friedman and Nissenbaum used 'systematic'; we made this change to emphasize that algorithmic bias often arises through unintentional oversights rather than requiring specific biased intents as 'systematic' implies. As a definition for data bias, we use Olteanu et al.'s 'a systemic distortion in the data that compromises its representativeness' (Olteanu, Castillo, Diaz, & Kiciman, 2016) as starting point. Note however that this raises an immediate dilemma: if data is completely representative of reality, it will also reflect the very real societal biases and existing disadvantages, and could potentially echo or amplify these societal biases. This means that 'biasing' the data against these biases may be important (Bolukbasi, Chang, Zou, Saligrama, & Kalai, 2016). We here supplement that definition with representativeness 'necessary for the application at hand' and perhaps 'representative' of the world teams would like to represent.

Frameworks and Types of Biases

Friedman and Nissenbaum present a taxonomy of biases in computational systems with top level categories of Preexisting Bias, Technical Bias, and Emergent Bias (Friedman & Nissenbaum, 1996). While Friedman and Nissenbaum's work was often prescient, it is difficult to use this taxonomy to address algorithmic and data bias issues in practice. Their categorization does not point to underlying causes, making it somewhat challenging to use the framework in a solutions oriented manner.

More recent taxonomies of algorithmic and data bias allow us to classify problems in a way that points out how to intervene and correct biases. The Baeza-Yates taxonomy consists of 6 types of bias: activity bias, data bias, sampling bias, algorithm bias, interface bias, and self-selection bias (Baeza-Yates, 2016). These biases form a directed cycle graph; each step feeds biased data into the next stage where additional and new biases are introduced. The cyclical nature of bias makes it difficult to discern where to intervene; models like Baeza-Yates' help break down the cycle and find likely targets for initial intervention.

Though biases exist and are propagated through all types of data, one of the most common types of data that practitioners currently use is social data. Social data encompasses content generated by users, relationships between those users, and application logs of user behaviors (Olteanu et al., 2016). The framework presented by Olteanu et al. comprehensively examines biases introduced at different levels of social data gathering and usage, including: user biases, societal biases, data processing biases, analysis biases, and biased interpretation of results.

Translation Into Bias Identification Processes

A major challenge is translating the growing, but scattered literature into a step-by-step process that works in practice. Unfortunately, in many cases the methods to assess, and certainly how to address a problem are not yet available.

The first step to correcting algorithmic biases is *identification* of *potential biases,* for which we have three possible entry points:

- Biases in input data
- Computational biases that may result from algorithm and team decisions.
- Outcome biases, for example for specific user groups (gender, age) or for specific domains (e.g. having really good recommendations in one genre over another).

Per definition, the first two categories can be done even before a project has been started, whereas the third category requires domain knowledge and at least a predictive model. Particularly challenging is that to be able to measure outcomes, we need to not only assess which facets would be important to explore, but also which evaluation metrics are actually valid - which is many cases can be very large projects themselves.

After the identification of potential issues, a prioritization has to be made of which of these issues are most pressing, and how to assess them. While eliciting bias targets from the bottom up is a positive initial route, it is still essential to prioritize which biases to tackle first. The problem is often not that identifying potential algorithmic biases is a difficult task; it is that looking for candidate algorithmic biases will surface a large number that it becomes difficult to determine which biases to tackle first and which are currently intractable and better suited as long term goals. Some bias targets are clearly long term, e.g. finding that a highly used metric loses much of its predictive power for subpopulations, while others may require simpler changes like modifying a data sampling paradigm for training models. Biases may compound and interact. For example, initial model training on a homogenous population may create an application that serves that population best; this may attract more users that fit the initial training population and compound bias by continuing to provide homogenous rather than diverse training data. It is essential that these bias targets are prioritized by evaluating impact on users and future compounding effects. Note that prioritizing simply on size of the affected population alone would lead to biases in itself, and that, on the other hand, slightly degraded user experience for a subpopulation may not be fixable or require effort best spent elsewhere. Weighing these bias targets against each other involves a complex decision involving level of harm, ubiquity of bias, and business driven priorities.

After assessment, very specific domain knowledge will be necessary to fix the bias at hand. Very promising projects exist focused on debiasing particular techniques, see (Bolukbasi et al., 2016) for debiasing word embeddings, but there is no guarantee that those methods will exist for your specific problem. In large settings, multiple issues may interact—and very pragmatic challenges can be encountered as well.

Domain Challenges

Every application will have different bias issues to assess and address. For example, voice interfaces are rapidly gaining popularity, but, unfortunately, voice interfaces may amplify bias due to their unique affordances. For example, voice interfaces may struggle with regional accents (Best, Shaw, & Clancy, 2013). Language dialects also may result in worse accuracy and voice recognition (Tatman, 2017). Even if dialects and accents were perfectly recognized by voice interfaces, these interfaces would still struggle to counteract biases using common solutions from other modalities. Recommender systems often suffer from popularity bias, meaning that popular content is recommended far more frequently than the long tail of less popular items (Abdollahpouri, Burke, & Mobasher, 2017). Solutions to enhancing discoverability of the long tail of content include increasing serendipity and novelty among recommendations (Vargas & Castells, 2011). Unfortunately, users are often trying to accomplish a task quickly by voice and listing 10 search results that include some popular, some novel, and some serendipitous results may degrade the user experience because of the time it takes to verbally list them. Therefore, this task of countering popularity bias may be much harder in voice where only one result is often returned. The voice realm may be challenging to properly correct biases in but that does not make the task impossible.

A major struggle with many types of bias research is understanding whether the metric differences measured are due to algorithmic/data bias or simply due to natural demographic variation (Mehrotra et al., 2017). Bias audits often require splitting the population sample in a way that we can measure metric differences across these samples but this action confounds itself because each sample may behave differently to begin with. Given this confounding challenge, a particularly effective way to measure and correct bias may be finding problems where a ground truth answer is available. Springer et al. examine the types of content that current voice interfaces underserve due to content characteristics (Springer & Cramer, 2018). For example, current voice interfaces often transcribe dialect speech into Standard American English; this can result in a user asking for a music track titled "You Da Baddest" and the voice interface transcribing and searching for "You're the baddest" which may not result in finding the intended track. These entity resolution difficulties fortunately mean that some form of ground truth is available; whether content can be accessed through an interface or not. With the availability of ground truth, we can tease apart the algorithmic bias from demographic differences and quickly identify ways to correct bias. However, there is no ground truth of human experience, nor behavior.

Every modality and every domain will require its own assessment methods and solutions. The challenge is to develop processes that are lightweight for teams to implement in order to create a more equitable product.

Pragmatic Challenges

In this section, we present a few examples of pragmatic challenges that may be encountered when attempting to mitigate data and algorithmic bias in an industry setting. First, value must be established to motivate the prioritization of reducing specific unfair biases in production systems. Next, the work be developed in a way that harmonizes with the engineering practice of rapid delivery. Finally, longer-term changes in engineering culture are necessary to address bias as early as possible.

Prioritizing Correcting Bias

Engineering teams abide by a carefully planned roadmap of deliverables, with much energy devoted to maintaining their current systems and pushing new features to product. Setting aside time to measure and correct bias has to compete with other pressing priorities. It becomes hard to prioritize such projects where it's unclear how to assess their impact. Methods are not yet available, and case studies from literature demonstrate the extensive effort and expertise necessary (Mehrotra et al., 2017) and are not easily translated into practice. Furthermore, in a situation where features built from imperfect data have already been surfaced in the product, making significant changes in the feature may be perceived as too risky. Characteristics of different datasets, models and intended counter-measures may interact in unexpected ways. Teasing apart their effects can be challenging. Framing such work in terms of business goals, such as improving performance across markets and improvement of quality, is a compelling argument for pursuing this work (compared with, for example, unspecified appeals that bias should be important).

Proposing Minimum Viable Products

Agile development is arguably the dominant approach to product development in startups. In an Agile-style environment, there is an emphasis on quick delivery of minimum viable products followed by continuous iteration. In order to translate research on bias to solutions in product, it is necessary to propose a minimal solution that can be delivered and then improved. For example, is it possible to move forward with solutions on narrow use cases or with imperfect measurements? Caution is required here, to prevent the minimum viable product from simply being accepted as the final product. Long-view thinking is also necessary, so that even as imperfect products are delivered quickly, there is still a path of iteration toward a more ideal solution. In larger companies, as datasets and APIs will be developed as services for other product teams, it becomes important to develop ways of documenting data characteristics in ways understandable beyond the direct team that developed these.

Addressing Technical Debt Via Cultural Changes

In the early stages of a company's development, the issue of scaling globally seems impossibly distant. In this scenario, teams may accumulate technical debt as a result of limited access to resources and data. For example, they may train models on themselves in the absence of user data, or quality evaluations may by necessity have to be adhoc, resulting in models that reflect the demographics or tastes of the developers. Even as the user base grows, models may be overfit to current users rather than performant a global market. When company growth reaches a point where global scaling becomes a priority, new perspectives and attitudes are necessary. Diversity in hiring becomes more important. Longer-term cultural change and education toward bias-awareness would also encourage engineers to design models and features with delivery to a global audience in mind, avoiding bias-related technical debt at the outset of the design process.

To make sure that processes and tools land in practice, they have to be lightweight, pragmatic and easy to communicate to a wide variety of teams.

Discussion

To assess and address algorithmic biases, teams need lightweight tools to make these processes their own, rather than calls to action from elsewhere. While examples in literature exist of very specific auditing projects, general auditing tools widely applicable to industry currently do not. Future tools should allow examination of both inputs and metrics across content and population segments. This sort of general tool would facilitate teams finding bias among their own products. In addition to this, we need to translate the growing literature into methods that are applicable across domains and easy to communicate, while still informative enough to be of help.

Actively involving teams on the ground in this process is absolutely crucial. Shared understanding within industries and sharing of developed methods and lessons learnt, combined with a bottom-up application of frameworks by teams themselves appears most fruitful. An expert researcher coming in to a new team with a model in hand to examine systems will surely identify potential biases. However, it would be difficult to understand the finer details and potential side-effects. Changing datasets can have unforeseen effects elsewhere, how infrastructures, services, data and different parts of applications interact can be hard to understand when not deeply involved in its development process. Prescribing specific methods from afar will not work. Ensuring that team-embedded data scientists and data engineers themselves have tools and easily accessible resources to understand what to look out for, would be more fruitful.

References

AAAI. AAAI/ACM Conference on AI, Ethics, and Society – February 2-3, 2018. New Orleans, USA. Retrieved November 2, 2017, from http://www.aies-conference.com/

Abdollahpouri, H., Burke, R., and Mobasher, B. 2017. Controlling Popularity Bias in Learning to Rank Recommendation. In *Proceedings of RecSys'17.* ACM Press. Forthcoming.

Baeza-Yates, R. 2016. Data and algorithmic bias in the web. In Proceedings of WebSci '16, 1-1. ACM.

ACM Press. https://doi.org/10.1145/2908131.2908135

Best, C. T., Shaw, J. A., & Clancy, E. 2013. Recognizing words across regional accents: the role of perceptual assimilation in lexical competition. In *Proceedings of INTERSPEECH* 2013, 2128-2132.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. 2016. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Proceedings of NIPS'16*, 4349–4357.

Crawford, K. Oct 17, 2017. Artificial Intelligence—With Very Real Biases. *Wall Street Journal*. Retrieved from https://www.wsj.com/articles/artificial-intelligencewith-very-real-biases-1508252717

Dopplick, R. Jan 14, 2017. New Statement on Algorithmic Transparency and Accountability by ACM U.S. Public Policy Council. Retrieved November 2, 2017, from https://techpolicy.acm.org/?p=6156

FAT*. 2017, August 5. FAT*. Retrieved November 2, 2017, from https://fatconference.org/

Friedman, B., and Nissenbaum, H. (1996). Bias in computer systems. *ACM TOIS*, *14*(3), 330–347.

Mehrotra, R., Anderson, A., Diaz, F., Sharma, A., Wallach, H., and Yilmaz, E. 2017. Auditing Search Engines for Differential Satisfaction Across Demographics. In *Proceedings of WWW'17,* Perth, Australia.

Olteanu, A., Castillo, C., Diaz, F., and Kiciman, E. 2016. Social Data: Biases, Methodological Pitfalls, and Ethical Boundaries. SSRN Scholarly Paper 2886526. Rochester, NY: Social Science Research Network.

Springer, A., and Cramer, A. 2018. “Play PRBLMS”: Identifying and Correcting Less Accessible Content in Voice Interfaces. In *Proceedings of CHI‘18*, Montréal, Canada: ACM Press. Forthcoming.

Tatman, R. 2017. Gender and Dialect Bias in YouTube’s Automatic Captions. In Proceedings *EACL‘17*, 53. Valencia, Spain.

Vargas, S., & Castells, P. 2011. Rank and Relevance in Novelty and Diversity Metrics for Recommender Systems. In *Proceedings of RecSys’11,* 109–116. New York, NY, USA: ACM Press.

What Are You Hiding? Algorithmic Transparency and User Perceptions

Aaron Springer, Steve Whittaker
University of California, Santa Cruz, 1156 High St., Santa Cruz, California, USA
{alspring, swhittak}@ucsc.edu

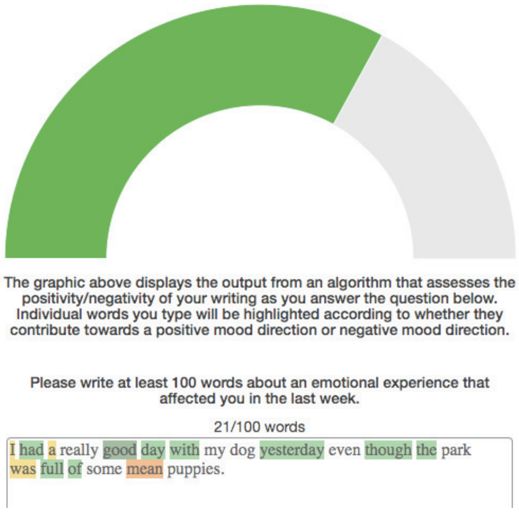
Abstract

Extensive recent media focus has been directed towards the dark side of intelligent systems, how algorithms can influence society negatively. Often, transparency is proposed as a solution or step in the right direction. Unfortunately, research is mixed on the impact of transparency on the user experience. We examine transparency in the context an interactive system that predicts positive/negative emotion from a users' written text. We unify seemingly this contradictory research under a single model. We show that transparency can negatively affect accuracy perceptions for users whose expectations were not violated by the system's prediction; however, transparency also limits the damage done when users' expectations are violated by system predictions.

Introduction

Intelligent systems powered by machine learning are pervasive in our everyday lives. These systems make decisions ranging from the mundane to the magnificent, from routes to work to recommendations about criminal recidivism. We, as humans, increasingly devolve more and more responsibility to these systems with little transparency or oversight. Concerns about how these systems are making decisions are building and this is only exacerbated by recent machine learning methods such as deep learning that are difficult to explain in human-comprehensible turns. These opaque systems have taken blame for major events such as the 2016 election of Donald Trump (Olson, 2016); they have been implicated in disadvantaging minority prisoners that are up for parole (Julia Angwin, 2016). Life-changing decisions are being made without any ability to examine the method or data these algorithms are using to make predictions.

This lack of transparency enables algorithmic problems to run amok. These systems have been shown to capture societal and human biases and perpetuate them systemically (Bolukbasi, Chang, Zou, Saligrama, & Kalai, 2016). Other work has indicated that a lack of transparency may lead users to accept output from algorithms that are simply random (Springer, Hollis, & Whittaker, 2017). It seems that we are seeing more automation bias than ever, (Cummings, 2004) we are increasingly willing to go along with what systems suggest rather than trying to critically examine those suggestions.



The graphic above displays the output from an algorithm that assesses the positivity/negativity of your writing as you answer the question below. Individual words you type will be highlighted according to whether they contribute towards a positive mood direction or negative mood direction.

Please write at least 100 words about an emotional experience that affected you in the last week.

21/100 words

I had a really good day with my dog yesterday even though the park was full of some mean puppies.

Figure 1. The E-meter System in the Transparent Condition after a user wrote about a positive experience and the E-meter predicted mood and associated words accurately

All of these problems have been met by calls for industry implementation of transparent algorithms and expanded research into issues of transparency and trust in algorithms. The response from industry has been anemic. Few commercial products have accepted this challenge of increased transparency. Yes, algorithms are evolving quickly, but the bigger issue seems to be that methods for transparency are not well understood. Results around the effects of algorithmic transparency have been mixed. Lim and Dey (Lim & Dey, 2011) found that increased transparency can make users question the algorithm when it's correct, therefore impairing the user experience. Users may also feel that these explanations simply cause additional processing without offering real value (Bunt, Lount, & Lauzon, 2012). On the other hand, transparency can help protect system trust by allowing users to understand why a prediction was made when that prediction violates their expectations (Kizilcec, 2016). It is difficult to form a coherent picture of the effects of transparency on the user experience from these conflicting results.

In this study, we explore the effects of transparency on the user experience. We experiment in the context of the E-meter, a system that predicts positive/negative emotion in a user's account of a past experience. The E-meter context allows us to examine how users interact with a system making predictions in an area the user is an expert in; only the user has the ground truth of their emotions. We make the E-meter more transparent to users to examine the conflicting nature of previous transparency research.

Specifically, we focus on how adding transparency to the E-meter influences user perceptions of accuracy. Accuracy is an important aspect of user experiences with intelligent systems. Users who believe that a system is accurate may be more likely to act upon its recommendations (Hollis et al., 2017). Therefore, transparency could play an important role in motivating user engagement with intelligent health and mental health applications. However, since recent research on transparency has mixed results, implementing transparency could also have net negative affects and push users away from using the application. We must unify these conflicting results in a way that illuminates a path forward for the use of transparency in intelligent systems.

Methods

Users

Users were recruited from Amazon Turk and paid $3.33 to evaluate the E-meter system. This evaluation took 13 minutes on average. We recruited 41 users to test the E-meter system across two conditions who were screened for stable mental health. Users were divided into 2 conditions: a control condition, and a transparent condition that allowed users to examine how each word affected the E-meter overall.

Machine Learning Model

Emotional valence predictions for users' experiences were predicted using a linear regression model trained on text from the EmotiCal project (Hollis et al., 2017). In EmotiCal, users wrote short textual entries and logged their overall mood, which gave us a supervised training set to train our linear regression on. We trained the linear regression on 6249 textual entries and mood scores from 164 EmotiCal users. Text features were stemmed using the Porter stemming algorithm (Porter, 1980) and then the top 600 unigrams were selected by f-score. Using a train/test split of 85/15 the linear regression tested at $R^2 = 0.25$; mean absolute error was .95 on the target variable (mood) scale of (-3,3). In order to implement this model on a larger range for the E-meter, we scaled the predictions to (0,100) to create a more continuous and variable experience for users. The mean absolute error of our model indicates that the E-meter will, on average, err by 15.83 points on a (0,100) scale for each user's mood prediction.

E-meter System

The E-meter (Figure 1) presented users with a web page showing a figure, a short description of the system, instructions, and a text box to write in. The system was described as an "algorithm that assesses the positivity/negativity of [their] writing". The instructions asked users to "Please write at least 100 words about an emotional experience that affected you in the last week."

As users wrote, the E-meter moved in accordance with the emotional valence of their writing; the meter could move positively, towards filling the gauge to the right, or negatively, towards emptying the gauge to the left, based on a regression model predicting the mood of their written experience. The E-meter was updated in real time after the user finished writing or removing a new word in the text box. The color of the E-meter changed depending on how positive or negative the overall rating, the E-meter changed from a deep red for very negative ratings, through orange, yellow, and light green, all the way to a dark green for very positive ratings of the user's' text.

The E-meter randomly assigned users to either a transparent or control condition. Those in the transparent condition were told that individual words would be highlighted to show the word's contribution to the E-meter's overall rating of their affect. In the transparent condition, users were able to see the extent to which each individual word they wrote contributed to the algorithm's evaluation of their emotional valence, using a method that highlighted words according to their evaluated affect. Users in the control condition did not see their words highlighted, though they could still see the movement of the meter after they finished writing each word.

We operationalized transparency in this space by passively highlighting the valence association of each word in the model because it offers an intuitive and persistent way to view how the E-meter responded. Essentially, this form of transparency offers a view directly through the text to the regression model powering the E-meter; it portrays how strongly the regression model correlates each word with positive or negative emotion. Offering this persistence of transparency allows users to reexamine what they had written when they finished and reconcile their overall E-meter rating with the fine-grained transparency from the text. When expectations are violated, users are prone to seek out more information to learn why the violation happened (Kizilcec, 2016). Our operationalization of transparency allows users to engage in this questioning mode.

	Coefficient	Std. Error	p-value
Intercept	6.991	0.377	**< 0.00001**
Expectation Violation	-1.736	0.601	**< 0.00001**
Transparency	-1.406	0.198	**0.007**
Transparency * Expectation Violation	1.057	0.441	**0.022**

Table 1. Linear Regression Predicting Users' Accuracy Perceptions

Survey

Following their experimentation with the E-meter, users were asked various questions about their experiences. Importantly, we asked users their perceptions of their own writing and their perception of the E-meter's rating of their writing on a 7-point Likert scale from "Strongly Negative" to "Strongly Positive". Users were additionally asked about the accuracy of the E-meter rating (7 point, "Very Inaccurate" to "Very Accurate") and how trustworthy they found the system (5 point, "Not at all" to "Extremely" trustworthy) and reasons for these ratings. The final questions were open-ended and asked about users' likes/dislikes and their' theories about how the system was calculating their final score.

Results

The majority of users across conditions found the E-meter to be "Accurate" or "Very Accurate" with the median being "Accurate". Users were slightly less trusting of the meter and found it to be "Moderately Trustworthy".

Transparency Moderates Expectation Violation

We calculate a user's expectation violation of their overall rating by subtracting the user's perception of their own writing (their expectation of the E-meter value if it were perfect) from the actual perception of the final E-meter score. If a user felt that their writing was "Strongly Negative" (1) but the E-meter rated it as "Slightly Negative" (3) then the user's expectation violation would be 2. Therefore, higher levels of expectation violation indicate that the user felt that the E-meter was less accurate overall while low levels of expectation violation should correlate with increased perceptions of accuracy. We refer to accuracy from the survey as holistic accuracy of the system, encompassing perceptions of the meter and the word highlighting. We see a strong relationship between expectation violation and accuracy in the control group, r = -.898, p < .00001 as well as in data from our previous study (Springer et al., 2017). Interestingly, this correlation between expectation violation and accuracy perceptions disappears in the transparent condition: r = -0.175, p = 0.488. We find that the relationship between expectation violation and accuracy perceptions is not so simple in the presence of transparency.

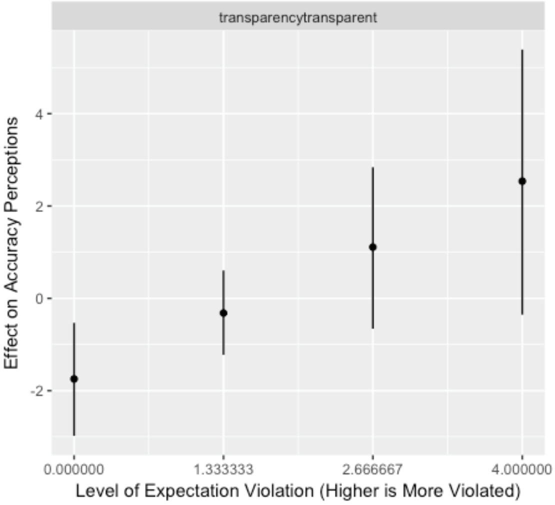


Figure 2. Transparency and Expectation Violation Interaction

We find that transparency and expectation violation interact complexly. We modeled this interaction using a linear regression predicting user accuracy perceptions (see Table 1). The regression was highly predictive R^2=.548, p < .000001. Transparency actually has a net negative effect on perceptions of accuracy. However, transparency begins to have a positive effect in the presence of increased expectation violation.

In the control condition expectation violation resulted in decreased perceptions of system accuracy. Users of the transparent system saw less decrease in accuracy perceptions as expectation violation increased. However, transparent system users have lower perceptions of the E-meter's accuracy even when their expectations are not violated.

Qualitative Examination of Anomalous Users

We now turn to qualitatively examining users to discern how transparency could cause this decrease in accuracy perceptions. We specifically examine these instances where the user stated that the E-meter was within 1 point of the user's own perception of their writing (on the 7-point scale from 'Strongly Negative' to 'Strongly Positive') and the user still rated the accuracy as inaccurate.

Lack of Personalization for Users

One problem concerned the generalized nature of the mood models. The models were trained on data consisting of 164 different users and thus the model learned general associations that hold across most people. Of course specific individuals may have completely different associations for specific words and making the machine learning transparent by highlighting these words can expose these differences more readily (Hollis et al., 2017). One user wrote: "Family is bad for me but it was marked in green." This user felt that his specific family experiences were highly negative in contrast with the model's association.

Questions Created by Transparency

The goal of highlighting words and making the E-meter calculation more transparent was to passively explain to users where their final positivity/negativity rating was coming from. However, for some users, this transparency just created more questions. One user noted that their final negative rating didn't make sense "because the rating did not correspond to the number of identified words". Other users wanted to know how the ratings the highlighted words were originally identified and their association with mood calculated. Care needs to be taken in the presentation of transparency to avoid damaging user perceptions by evoking unneeded questions.

Conclusion

Algorithmic transparency has a complex relationship with user perceptions of algorithmic accuracy. In our experiment, transparency effectively compressed user perceptions of accuracy. Users with the most violated expectations had better perceptions of the E-meter's accuracy compared to their control counterparts. However, users in the transparent condition were less likely to regard the E-meter as highly accurate even when it did not violate their expectations at all. This result unifies previous, seemingly contradictory, results that indicated that transparency had positive or negative effects (Kizilcec, 2016; Lim & Dey, 2011).

Whether or not transparency is a net positive in an application may depend on other characteristics of the application. For example, if an intelligent system is highly accurate in its predictions, then increasing the transparency of the application may have a net negative effect of slightly lowering perceptions of accuracy. If an application is often inaccurate, then transparency could have a net positive by tempering those negative perceptions of accuracy. Of course, other factors can influence whether transparency is needed, such as the impact of the decision that the algorithm is influencing.

In addition, we find a few routes through which transparency can decrease perceptions of accuracy. Transparency exposes the fact that these models are general and learned from a societal space that embodies many correlations that may not be personally relevant to each user. These correlations can be learned over time, like in EmotiCal (Hollis et al., 2017), but on first interaction this is a difficult problem to solve. Possible solutions could include "emotional onboarding" analogous to what recommender systems use to solve the cold start problem, others may include scanning a user's social media and building a profile for them through the given information (Warshaw et al., 2015). Another problem stemmed from the exposed information from transparency creating additional questions which led to users doubting system accuracy. Overall, we want a seamful design allowing users to explore the model but only to the point that is helpful.

Rather than simply focusing on how to present transparency in ways that don't evoke more questions of the algorithm, researchers should instead focus on recognizing expectation violation. If a system can recognize in real time when a users' expectations of the system were violated, it can choose to be selectively transparent. Selective transparency would maintain the positive aspects of transparency without making users unduly question the application.

References

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In *Advances in Neural Information Processing Systems* (pp. 4349–4357).

Bunt, A., Lount, M., & Lauzon, C. (2012). Are explanations always important?: a study of deployed, low-cost intelligent interactive systems. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces* (pp. 169–178). ACM.

Cummings, M. L. (2004). Automation bias in intelligent time critical decision support systems. In *AIAA 1st Intelligent Systems Technical Conference* (Vol. 2, pp. 557–562). AIAA.

Hollis, V., Konrad, A., Springer, A., Antoun, C., Antoun, M., Martin, R., & Whittaker, S. (2017). What Does All This Data Mean for My Future Mood? Actionable Analytics and Targeted Reflection for Emotional Well-Being. *Human–Computer Interaction*. https://doi.org/10.1080/07370024.2016.1277724

Julia Angwin, J. L. (2016, May 23). Machine Bias [text/html]. Retrieved October 27, 2017, from https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

Kizilcec, R. F. (2016). How Much Information?: Effects of Transparency on Trust in an Algorithmic Interface (pp. 2390–2395). ACM Press. https://doi.org/10.1145/2858036.2858402

Lim, B. Y., & Dey, A. K. (2011). Investigating intelligibility for uncertain context-aware applications. In *Proceedings of the 13th* *international conference on Ubiquitous computing* (pp. 415–424). ACM. Retrieved from http://dl.acm.org/citation.cfm?id=2030168

Olson, P. (2016, November 9). How Facebook Helped Donald Trump Become President. Retrieved October 27, 2017, from https://www.forbes.com/sites/parmyolson/2016/11/09/how-facebook-helped-donald-trump-become-president/

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *14*(3), 130–137. https://doi.org/10.1108/eb046814

Springer, A., Hollis, V., & Whittaker, S. (2017). Dice in the Black Box: User Experiences with an Inscrutable Algorithm. In *2017 AAAI Spring Symposium Series*.

Warshaw, J., Matthews, T., Whittaker, S., Kau, C., Bengualid, M., & Smith, B. A. (2015). Can an Algorithm Know the "Real You"?: Understanding People's Reactions to Hyper-personal Analytics Systems (pp. 797–806). ACM Press. https://doi.org/10.1145/2702123.2702274

Hey Scout: Designing a Browser-Based Voice Assistant

Janice Y. Tsai, Jofish Kaye
Mozilla
janice@mozilla.com, jkaye@mozilla.com

Abstract

We present Scout, an open-source, browser-based voice assistant built into Firefox. We used an HCI-driven research-based approach with a focus on understanding how people are using commercially widespread dedicated device-based voice assistants (Alexa, Google Home) and the impact of different modalities and form factors (standalone device, mobile, laptop) on the user experience.

Introduction

Voice input will be an important part of how people will interact with technology for the foreseeable future. The question of feasibility for speech communication has evolved from science fiction to voice assistants that are a widely adopted, commercial success. These successful assistants have been brought to us primarily by large technology companies and have a variety of form factors, ranging from standalone devices (Amazon's Alexa, Google Home), to mobile phone and desktop-based agents (Microsoft's Cortana, Apple's Siri). Building on the success of these agents, we created Scout, a browser-based voice assistant for Mozilla.

We embarked on an HCI-driven research-based approach to product and feature definition and user experience (UX) design. In this work, we placed an emphasis on the UX, rather than the AI or machine-learning aspects of voice assistants, as these aspects are still relatively invisible to users. This data-driven agenda included the log analysis of Alexa History; surveys, interviews, and focus groups to understand the use of existing voice assistants, their failings, and the feature wishes of both users and non-users; and the deployment of an alpha product to understand the use of Scout in the wild.

Motivation

Mozilla's mission focuses on creating open and accessible Web technologies. One aspect characterizing the existing voice assistant market is that of the race to create platform-specific "incompatible proprietary fortress[es]" (Rosenberg 2017). We created Scout as an open-source voice assistant built into the Firefox browser.

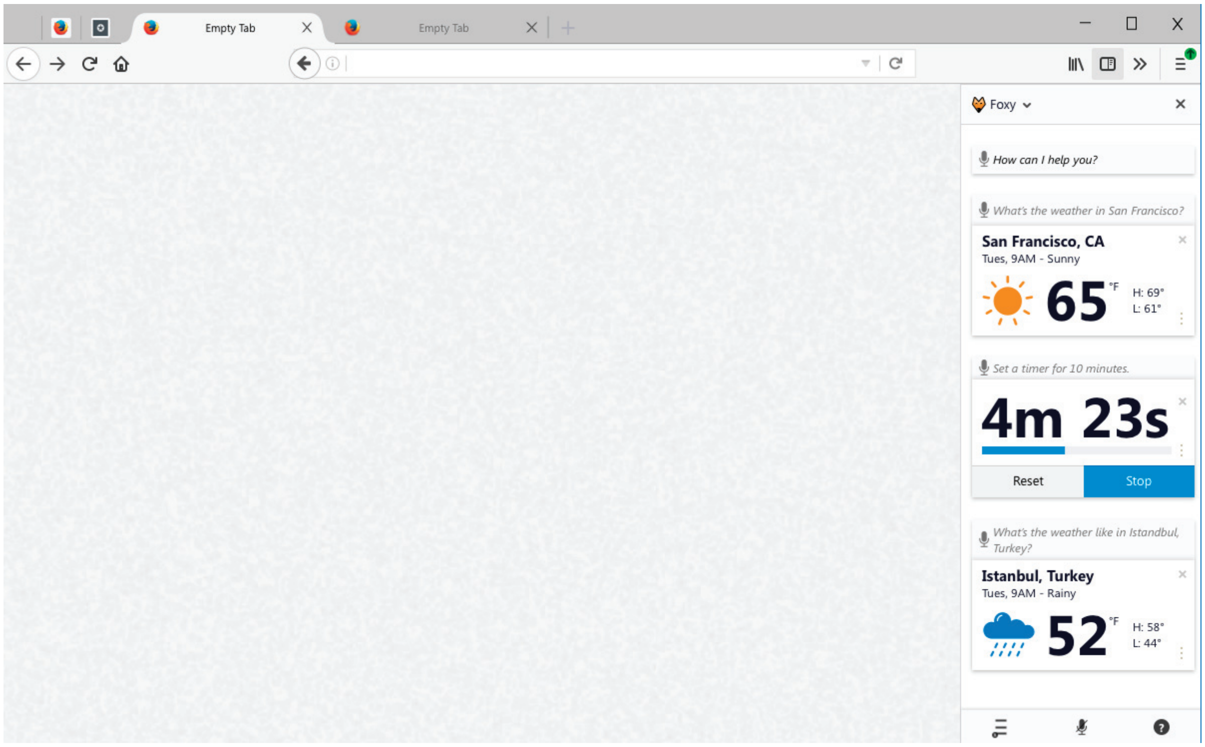
Background

Natural, voice-based interaction with computing systems is facilitated by spoken dialogue systems (comprised of speech technologies, language processing, dialogue modeling) (McTear 2002) with a focus on "humanlike" behavior (Vassallo et al. 2010). Three high level design principles for the functionality of voice assistants center on goals, specifically related to *tasks* (accomplishing things), *conversation* (facilitating communication and understanding), and *relationships* (maintaining connection and influence) (Shechtman and Horowith 2003). We explore these areas as we set about defining a browser-based voice assistant.

Research

We focused on Amazon's Alexa to understand the tasks required of and the conversation with a voice assistant. In July and August, 2017, we collected the Alexa History logs of 82 participants with a total of 193,665 commands (participant mean: 2,176 commands). Our participants owned 147 Alexa devices (mean: 1.79, median: 1, mode: 1) primarily in households with other people (82.9%), with an average of 2.82 people per household. The average age of the primary Alexa account holder was 31.9 years old, and 41.5% of the households had children under 18 years old.

People primarily used Alexa to play music (33%), to interact with their IoT devices (15%), and to conduct general information queries (14.5%). This information informed the development of the features and capabilities available in Scout.



How can I help you?

What's the weather in San Francisco?

San Francisco, CA
Tues, 9AM - Sunny
65 °F H: 69° L: 61°

Set a timer for 10 minutes.

4m 23s

What's the weather like in Istandbul, Turkey?

Istanbul, Turkey
Tues, 9AM - Rainy
52 °F H: 58° L: 44°

Sidebar-based design for a browser-based voice assistant.

User Experience

Building a voice assistant into Firefox afforded us the ability to have a visual user interface. We implemented this in the sidebar of the browser. This visual space allowed us to implement "cards" to provide a response to the user. (At this time, Scout is unable to speak back to the user.) We are focused on task goals, and will continue to improve the user experience as continue to conduct research to understand the conversational and relationship-based goals that users want with a voice assistant.

Architecture

The Scout voice assistant is a web extension that runs inside of Firefox. It allows you to interact with it via voice and a browser-based extension sidebar. It uses Snowboy,[1] a customizable wake word detection engine to allow users to create a personal model for wake-word detection.

The Demo

The demo utilizes a laptop with a Google Slides presentation open in Firefox. The presenter walks up to the podium with his hands in his pockets and proceeds to showcase the functionality of the Scout voice assistant. This functionality includes the following:

- Playing music
- Conducting searches
- Navigating slides
- Setting timers and more.

Conclusion

With the success and availability of voice assistants, a new canvas of research around the interactions, user experiences, and design paradigms for artificial intelligence is now available. We are creating Scout, a browser-based voice assistant, to explore the use and impact of voice interaction in a web-based environment.

Bios

Joseph 'Jofish' Kaye is a principle research scientist working in the Mozilla Emerging Technologies team. He uses a variety of methods, including big data and qualities research to understand user needs and practices in the HCI space.

Janice Y. Tsai is a senior research scientist on the Mozilla Emerging Technologies team. Her research interests are in usable privacy and public policy.

[1] https://snowboy.kitt.ai/

References

McTear.M. 2002. Spoken Dialogue Technology: Enabling the Conversational User Interface. *ACM Comput. Surv.* 34, 1: 90–169.

Rosenberg, S. 2017. “Voice Assistants Aren’t So Easy to Fire.” Wired, Oct. 11, 2017. Accessed Oct. 26, 2017. https://www.wired.com/story/voice-assistants-arent-so-easy-to-fire/.

Shechtman, N. and Horowitz, L. 2003. Media Inequality in Conversation: How People Behave Differently when Interacting with Computers and People. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (CHI ’03), 281–288.

Vassallo, G., Pilato, G., Augello, A., and Gaglio, S. 2010. Phase Coherence in Conceptual Spaces for Conversational Agents. In *Semantic Computing.* Hoboken, NJ: John Wiley & Sons, Inc.

Committee of Infrastructure: Civic Agency and Representation

Jason Wong
Media Design Practices
ArtCenter College of Design
Pasadena, CA, USA
jwong30@inside.artcenter.edu

Abstract

This article examines bias exhibited through machine vision, over optimization in machine learning, representation for previously unrepresented stakeholders, and agency within the context of a speculative city council meeting. In this paper, I present a project that purposely shows bias in order to reveal how easily machine learning algorithms can problematize a situation. I also have created representatives in the form of Artificial Intelligent systems for both human and non-human communities. I identified a scenario, voting for the removal of traffic lights, as a medium to discuss the topic of bias and representation. This work contributes to the discourses of speculative design and civics in design research.

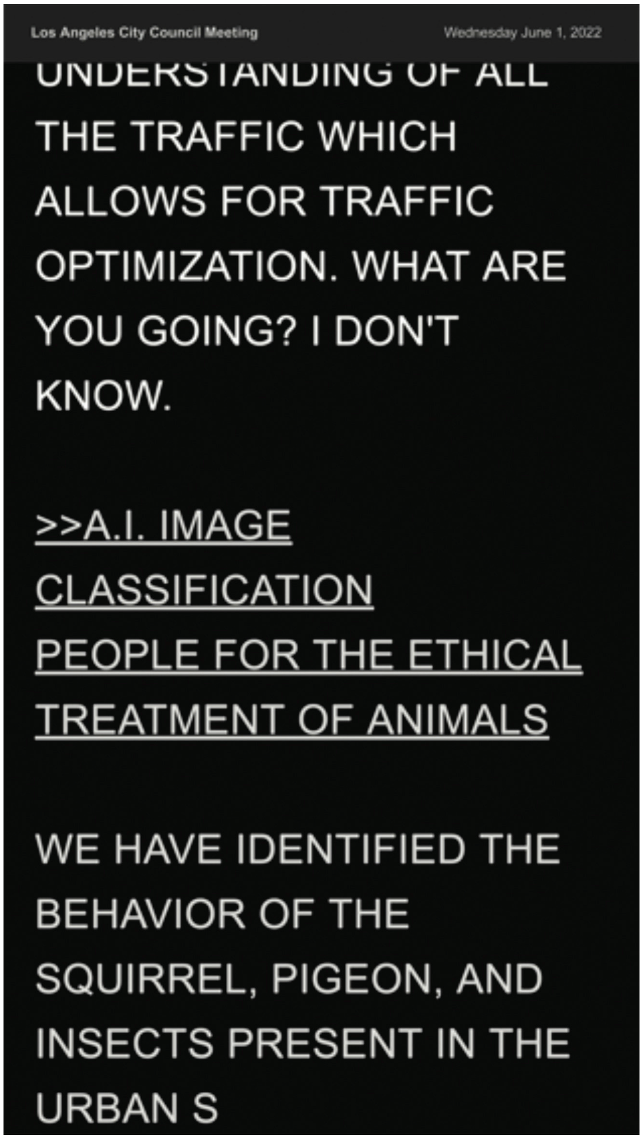
Introduction

Artificial intelligence's nascent dominance and pervasiveness in the mediation of our everyday interactions is making it harder to create boundaries of personal space and to distinguish how ruthlessly it changes human behavior. Individuals are becoming increasingly reliant on these services as they become more embedded and necessary to daily life. These Artificial Intelligent (AI) systems such as personal assistants, route planning, language translation, and image sorting are creating a multi-layered and interdependent environment that obfuscates the machinery of the system.

Machine Learning (ML) algorithms in the civic sector are currently being used in the criminal justice, health, and welfare sectors to augment the decision making of federal and state employees. The use of algorithms in the criminal justice system this has led to sentencing imbalances amongst different races. The COMPAS algorithm used by the Department of Corrections in Wisconsin, New York, and Florida has led to harsher sentencing towards African Americans (Angwin et al. 2016). Although we cannot ascertain the specific type of algorithm used in the criminal sentencing sector, the same issue extends to ML. Bias, if left unexamined, exacerbates the problems it is trying to mediate.

Committee of Infrastructure Design Project

Committee of Infrastructure is a speculative design project (Dunne and Raby 2013) that interrogates the issue of agency and representation with the domain of ML and AI. The project considers how humans and AI systems interact with each other in a local government setting to negotiate issues pertaining to a local community. It explicitly positions human representatives and AI representatives as stakeholders within a local council meeting. These two types of representatives express conflicting positions, ideologies, and motivations. An AI representative through the use of ML can now understand the behavior and represent a community that has not been previously recognized such as animals living in the urban environment. The project asks whether our AI civic representatives will be as intolerant as humans, or can we program a diversity of voices and positions to reflect the populace and other forms of life that create our world? The meeting proposes a platform that allows both human and nonhuman entities to be considered as meaningful representatives of a particular position. Like a real city council meeting, the projects intends that these speculative stakeholders will be able to implement changes to their local political system. The project seeks to engage with the nascent field of speculative civics (DiSalvo, Jenkins, and Lodato 2016). In order to convey the extent of the project, Committee of Infrastructure includes a transcript, meeting notes, photo manipulation, video, and infographics.



Los Angeles City Council Meeting Wednesday June 1, 2022

UNDERSTANDING OF ALL THE TRAFFIC WHICH ALLOWS FOR TRAFFIC OPTIMIZATION. WHAT ARE YOU GOING? I DON'T KNOW.

>>A.I. IMAGE CLASSIFICATION PEOPLE FOR THE ETHICAL TREATMENT OF ANIMALS

WE HAVE IDENTIFIED THE BEHAVIOR OF THE SQUIRREL, PIGEON, AND INSECTS PRESENT IN THE URBAN S

Figure 1:Transcript Excerpt

Speculative City Council Meeting

Situating the project in the Los Angeles Echo Park neighborhood provides further context for the council meeting. The meeting is proposed to take place in 2023. The issue discussed amongst stakeholders is a ballot measure to remove all traffic lights at the intersection of Sunset Boulevard and North Alvarado Street in order to create a fully autonomous intersection. Autonomous cars will sense objects, things, and people through machine vision and proximity detection. These autonomous cars will communicate with different Industrial Internet of Things (IoT) such as smart streetlights and speed sensors to avoid collisions and efficiently move through traffic. Not only will pedestrians have their presence notated by machine vision from smart streetlights but also from embedded sensors on clothing such as a magnetometer, GPS, gyroscope, accelerometer, and proximity sensor. These sensors will allow open communication between pedestrians, cyclists, and autonomous vehicles.

With the advent of machine vision and sophisticated statistical models, AI systems and human representatives will able to speak on behalf of new groups. Machine vision allows for detection of non-human living beings such as birds, insects, dogs, etc. Machine vision would be utilized to understand their specific behavior on the streetscape (e.g. location and duration) and protect them from becoming injured by autonomous vehicles. In this scenario engineers and scientists can predict the location of animals in near real time allowing for communication amongst all other forms of traffic. In turn both human and AI representatives will advocate on their behalf as data can support their arguments.

A city council meeting transcript provided the framework to work within, wherein each participant constructs arguments representing the the interest of their respective organizations. The four groups in the meeting are the city council members, People for The Ethical Treatment of Animals (PETA), L.A. Department of Transportation (DOT), and the Alliance for Biking and Walking. Stakeholders include engineers, presidents, AI experts, Smart Roads, and sensors. The transcript was created by using the Karpathy char-rnn machine learning algorithm[1] that used seminal texts important to the ethos of each agency or the technical jargon required to speak as an expert. For example the L.A. DOT representative learned to speak from City of Los Angeles Transportation Impact Study Guidelines[2] and Traffic Studies Policy and Procedures[3]. Once given instructions about the content of the arguments the algorithm creates a wholly artificial language mimicking the mechanics of a Los Angeles city council meeting. The constructed language is absurd and awkward, but exhibits AI systems and humans conversing amongst themselves.

[1] https://github.com/karpathy/char-rnn

[2] http://ladot.lacity.org/sites/g/files/wph266/f/COLA-TISGuidelines-010517.pdf

[3] https://planning.lacity.org/eir/8150Sunset/References/4.J.%20Transportation%20and%20Circulation/TRAF.03_LADOT%20Policies%20and%20Procedures_2013.pdf

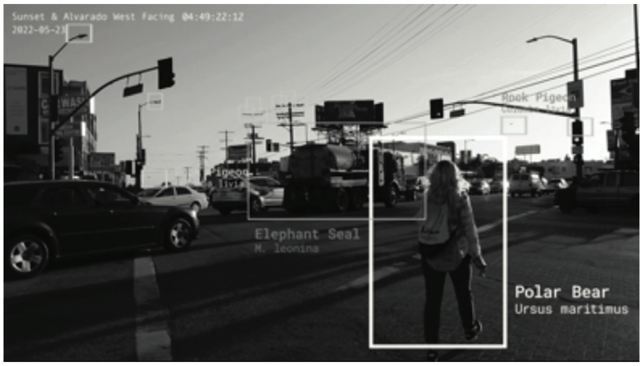


Figure 2: P.E.T.A. Image Classification

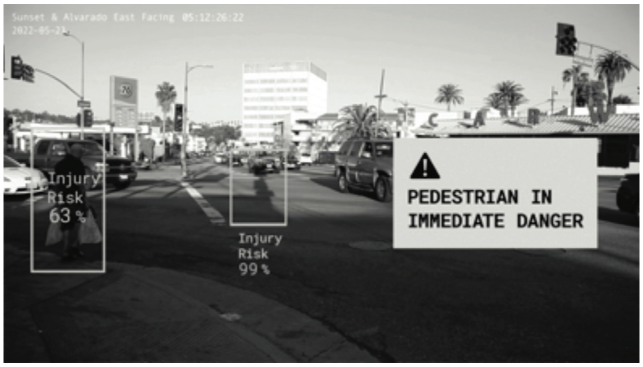


Figure 3: Alliance for Biking and Walking Image Classification

Machine Vision Bias and Over Optimization

In addition to the verbal arguments of each organization, sets of video evidence (machine vision) display each organization's motivations. Specifically, computers classify static and moving objects within a video. By classifying these objects and assigning value to each object a hierarchy is created allowing communication between vehicles, people, and animals. As one organization focuses on traffic efficiency, another focuses on pedestrian injury risk, and the remaining organization focuses on animals. Each organization over optimizes their object classification system to produce supporting analytics that promote their motivations. For example PETA mistakenly classifies every moving object in the video as an animal. The videos reveal the bias coded into object classification. Not only is the bias present, but also the ruthless efficiency of the machine vision system doesn't allow for flexibility. In effect, its extreme sensitivity to detect all stimuli creates errors.

More broadly, AI systems are subject to the same fallibility that is present in the day to day interactions between humans. Therefore, as we continue to rely more on these AI systems we must be aware of how they can lead us astray. We cannot blindly follow the decisions made by AI systems. We must challenge them when they are wrong, assess what is missing, and be inclusive of a broader set of individuals and other forms of life. This process is ongoing and must be constantly revisited and updated to reflect the constant flux of society and culture. Humanity and the larger world's present and future coexistence with technology is reliant on the delicate balance of us and AI systems. Creating an open process that informs the populace and that is inclusive is necessary. Committee of Infrastructure proposes the model of civic dialogue as a framework to interact with AI systems.

Conclusion

In this paper, I presented the Committee of Infrastructure as a speculative design project. The purpose of the project is to provoke and create a vehicle to discuss ML bias, over optimization, conflicting positions, overlooked considerations, and data classification. Additionally, AI systems have the potential to become representatives for human and non-human communities. These systems are privy to the same biases that humans have as they have been created by humans. Moreover, these AI systems can become over optimized to perceive the world in a specific way. By explicitly revealing their shortcomings I hope to demonstrate how AI systems are not to be blindly trusted, but should be subject the same form of scrutiny as a bill or a law. Potentially, programmers of AI systems could create a dossier of motivations, origin, and data sources visible to programmers and users. A record could be used to contend specific points that affect the outcomes of an AI system. By detailing all the variables a discussion amongst a broader set of stakeholders should challenge preexisting assumptions and provide evidence to throughly negotiate how these systems influence daily life.

The project is not intended to be a realized depiction of a city council meeting. However, by using speculative design the project can consider a future civic scenario that is not bound to technical limitations of a working prototype. The project allows experts in the AI field, designers, and policymakers to create a discourse about the potential effects of AI systems in the civic space. I believe that including these different experts is a necessity to create an approach that doesn't exclude and provokes new methodologies to address the ethical and moral uncertainties in artificial intelligence.

Acknowledgments

The author would like to acknowledge Phil Van Allen and Ben Hooker for their feedback and direction throughout the project.

The project was completed at Art Center College of Design as part of module examining the IoT within the context of the neighborhood.

References

Angwin, J.; Larson, J.; Mattu, S.; and Kirchner, L. 2016. *Machine Bias*. ProPublica. https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing

DiSalvo, C.; Jenkins, T.; and Lodato, T. 2016. Designing Speculative Civics. In P*roceedings of the 2016 CHI Conference on Human Factors in Computing Systems*: 4979-4990. San Jose, Calif.: CHI.

Dunne, A., and Raby, F., 2013. *Speculative Everything: Design, Fiction, and Social Dreaming*. Cambridge, Mass.: MIT Press

Machine Learning as a UX Design Material: How Can We Imagine Beyond Automation, Recommenders, and Reminders?

Qian Yang
yangqian@cmu.edu
Human-Computer Interaction Institute, Carnegie Mellon University
Pittsburgh, Pennsylvania

Abstract

I design user experiences for machine learning (ML) applications across several application domains, from clinical decision support systems to context-aware mobile services, to autonomous cars. In this paper, I share three cases in which I attempted to leverage ML as a design material, envisioning new forms and new purposes for this technology. I reflect on the challenges encountered and the lessons learned. On reflection, I realized that many of the challenges are not unique to particular ML problems or designers, but in the inherent tension between user-centered design and data-driven design. I hope to initiate a reflective discussion on these overarching challenges in designing ML and to highlight the opportunities in addressing them systematically.

Introduction

Machine learning (ML) increasingly plays an important role in shaping how users interact and experience technology products. From mundane spam filters to personalized news feeds to conversational agents, it can seem like ML is in almost every new technology product and service. Both UX practitioners and researchers have noted this trend and have become especially interested in design opportunities surrounding ML.

However, recent studies show that UX practitioners seem unprepared to effectively leverage ML capabilities. They struggle to understand the capabilities and limitations of ML in the context of their designs, even though many of them understand how ML works generally. Also, they typically join projects near the end, after the functional decisions have been made. "*Design teams are simply putting lipstick on the pig*" (Dove et al. 2017). Researchers noted two symptoms of these struggles of designing ML: One is that designers often fail to notice obvious places where ML could improve UX (Yang et al. 2016b). The other is a lack of design innovation in ML: "*Designers often have cliched understandings of this medium, driven by the hype or criticism surrounding the field. For example, some typical stereotypes are that wearables are for fitness, artificial intelligence is for automating tasks.*" (Allen and Hooker 2017)

Recently, HCI/design researchers began taking actions to address this problem. Some focused on teaching the technical concepts of ML to designers (Hebron 2016). Others organized workshops, bringing together groups of artists, designers, and technologists to collectively explore how ML might function as a creative material (Gillies et al. 2016; Kuniavsky, Churchill, and Steenson 2017). This emergent strand of work within HCI has spawned the notion of *ML as a design material* (Dove et al. 2017; Yang et al. 2016b).

My work is a part of this growing area of inquiry. We designed ML systems across several application domains, ranging from clinical decision support systems, to adaptive mobile user interfaces (UIs), to driving style of autonomous cars. These experience have led us to consider the unique challenges of designing ML applications, especially when new designs arose as a part of our user-centered design intention, rather than from an available dataset. To advance on these challenges, we have since worked to better understand ML as a design material. We interviewed experienced designers who have been designing ML systems for more than a decade, to probe the current best practice in the industry (Yang et al. 2018). We analyzed more than 2,400 HCI research publications, mapping out the design opportunities ML's technical advances have made available (Yang, Banovic, and Zimmerman 2018).

In this paper, I share three cases in which I attempted to leverage ML as a design material across multiple application domains. I reflect on the challenges encountered and the lessons learned. My goal is to demonstrate that many of the challenges are not unique to particular ML problems or designers, but are a result of the inherent tension between user-centered design and data-driven design. Through this paper, I hope to initiate an reflective discussion on these overarching challenges in designing ML, and to highlight the opportunities in addressing them systematically.

Machine Learning as a Design Material

While most recent papers discuss challenges, let me start by articulating this vision of *ML as a design material*.

Technology as Design Material

When I am talking about *ML as a design material*, I am talking about its design innovation. Louridas notes the difference between technical and design innovation when describing the difference between designers and engineers. He notes that engineers create new technology that allow new capabilities. Designers, he claims, do not invent new technologies. Instead, they create novel and valuable assemblies of known technologies (Louridas 1999).

When taking a technology as a design material, designers first develop a *tacit* understanding of how the technology opens up and constraint design possibilities. They then innovate by engaging in reflective conversations with design materials, envisioning things that have never before existed (Schön 1984; Wiberg 2014). Schön describes how designers reflect in action, how they conceive of what they want to make while in the act of making it (Schön and Bennett 1996). Gaining a deep understanding of how designers engage in a "conversation with their materials" and how the materials "talk back to the designer" is an important component in understanding design and design practice, and an important means of sharing design knowledge across disciplines (Zimmerman, Stolterman, and Forlizzi 2010).

Machine Learning as Design Material: A Vision

The last few years witnessed explosive technical advances in ML. Our review of literature shows that a total of 1,939 HCI publications mentioned "machine learning" since the publication of the first mention in 1969; More than half of the papers addressing ML were published in the past five years. The overwhelming majority of papers appeared in venues with a technical focus (Yang, Banovic, and Zimmerman 2018). This fast growing set of technical capabilities offer exciting new materials for designers.

Technical advances are often followed by design innovation, where designers envision many new product forms that apply the technical advance into different aspects of peoples lives.

For example, the 1962 cassette record represented a technical advance, providing an easier way to make audio recordings. This technical advance was followed by a wave of design innovation. Designers created new forms including home stereo systems, boom boxes, personal players such as the Walkman, automotive tape players, duo-cassette player-recorders that better supported making mix tapes, dictation devices, and phone answering machines.

Another example is haptics. Moussette has an interest in innovating haptics. He sketched with this technology, setting a goal for himself to make a new haptic device each day for several weeks. His work then produced a new language for talking about the aesthetics of haptics, as well as several simple devices that expose haptics' design possibilities beyond a buzzing phone (Moussette 2012).

We have not yet seen a similar type of design innovation taking place with ML; where a technical advance is followed by many new product forms. "*Today, it seems that ML systems are as creative and interesting as the data scientists that make them*." (Dove et al. 2017) This marks a ready opportunity for design innovation. By engaging with and understanding this technology and its continued advances, designers can envision new forms and new purposes for this technology, and radically re-imagine what it might be or might do.

Motivated by this vision, researchers and designers have began to investigate *ML as a design material*. This strand of work is quite different from technical HCI research which typically utilizes ML as a tool to extend or accelerate well established interaction forms. This is also quite different from other lines of design inquiry associated with ML in which designers often join ML development after the functional decisions have been made.

Case Studies

As part of the growing area of inquiry on *ML as a design material*, my work envisions new products that fit ML's technical advances into different aspects of peoples lives. Of these I draw three cases; I reflect on the challenges encountered and discuss how each of the cases advanced my understanding of ML from a UX design perspective.

Adding Machine Learning to Existing UX Design

The first case is drawn on a simple project in which we extend an existing mobile application with a ML feature that reduces navigation and selection efforts.

> *My collaborators and I attempted to enhance our mobile transit app, Tiramisu, with ML-powered adaptive interfaces. The new ML feature is intended to anticipate users menu-item selections, reducing their navigation and selection efforts. Yet we encountered two problems that made this impossible. First, we had not logged the information needed to infer what users most likely wanted to do. Second, we had not properly motivated users to provide good labels that would support adaptation.*
>
> *On reflection, we realized UX designers should identify and refine UI adaptions when sketching wireframes. Although challenging, recognizing learning opportunities when wireframing will be an important goal for future IxD practices.*
>
> *To advance on this challenge, we developed a set of UI design patterns that define where and how ML adaptations might be applied. We also created a new form of wireframe that support UIs that change over time, across use contexts. On this modified wireframe, designers can annotate pre-adapted and adapted interaction paths, and document the data needed to make an inference, a recovery method in the case of an inference error, and an inference quality needed to trigger adaptation.* (Yang et al. 2016b)

This project's design problem, adaptive UI, and its underlying ML model are simplistic in comparison to the ML problems HCI research focuses on today. Yet we encountered unanticipated barriers in this simple project. This is because we had not yet taken ML into consideration at the early stages of the design process.

We learned two lessons. First, understanding how ML works does not sensitize designers to the opportunities to apply ML in their designs. Most researchers on the above project had many years of experience designing, developing, and evaluating ML systems. Yet we still failed to anticipate simple ML opportunities in our design. This is partially because ML is not part of a typical user-centered design process; wireframing tools or patterns do not yet support the UIs that change over time or personalize to users.

There is a clear need for design research that helps expand designers perceived application of ML, and sensitizes them to the breadth of its design possibilities. In our conversations and workshops with UX practitioners, we noticed that they often failed to recognize many ML applications that they use every day, especially the ones that have so successfully adapted to their interactions that they have become invisible or unremarkable. Instead, designers only referenced a few classic, and somewhat failed, designs exemplars (i.e. Clippy, Tay, email spam filter) (Yang et al. 2018). Design researchers could create sensitizing concepts to communicate ML design opportunities that are not instantly recognizable.

The second lesson is that UX cannot not be an afterthought of ML. The lack of planning for ML could stand as a barrier to operationalizing this technology in current product development processes. The software development community has learned over many years that usability and user experience should be considered early in the development process and not added as an afterthought at the end. HCI has since developed best practices and tools that scaffold better software development processes. We see a similar need in scaffolding ML considerations in UX design.

In our conversation with designers who regularly work on ML systems, we noticed that they "plan for ML" by working with telemetry data on day-to-day basis. In doing so, they searched for user behaviour patterns that ML can potentially leverage (Yang et al. 2018). Many taught themselves to create data visualization tools to capture "rich and compelling user stories", such that the quantitative analysis of user behavior "*do not privilege data scientists*". The designers thus were able to raise new ideas in enhancing their product with ML.

Adding UX to Existing Machine Learning System

The second case is a project in which we design a clinical ML-driven decision support system by applying a classic user-centered design (UCD) workflow (Council 2005).

> *A few years ago, a team of Bioengineering researchers approached us to design a ML system that helps cardiologists better decide whether and when to implant an end-stage heart failure patient with a mechanical heart pump. Like almost all other clinical decision support tools, the system at the time took a prototypical form: It takes in a list of patient condition measures and produces an individualized prediction of patient trajectory, including likely survival and other post-surgical risks.*
>
> *Taking UCD as our starting point, we first interviewed and observed clinicians caring for candidate patients at three different implant centers. Interestingly, our findings revealed that for most cases, clinicians do not find the implant decisions challenging, and thus would not likely engage with a ML system to aid with the decision. Instead, they would value the support for emergent cases, when they have very little data to predict how a critical patient might respond to available therapies. In addition, the implant clinicians would value a ML system that help upstream clinics and hospitals to refer patients to implant centers in time, before patients became too sick for an implant surgery.* (Yang et al. 2016a)
>
> *Since then, we have been working to develop a new ML system that infers implant outcomes of critical patients in intensive care units (ICU). We also have worked to obtain electronic medical records from local and primary care healthcare systems, in order to infer optimal referral windows for each patient. However, the training datasets for such systems are in their nature extremely difficult to curate and label. We do not yet know whether these two design solutions are technically achievable.*

This case illustrates a classic UCD process, from field study to sketching to prototyping (Council 2005). It was successful by many measures, especially through the lens of taking ML as a design material: We actively examined user needs, wants and social contexts; We envisioned new designs that seem technologically feasible, and users are likely to find valuable. However, we encountered significant barriers in prototyping and implementation. The technical and cultural validity of our designs remained unknown until the ML system was built and implemented.

It seems that ML challenges the general idea of prototyping; of making just enough of a system to assess if this is the right direction to go. ML seems to require a much higher level of commitment, requiring an unwieldy amount of data to create a functional prototype. This could conflict with UX mantras like "fail fast, fail often." (Dove et al. 2017) Consequently, in research, it is difficult to experiment with many different design solutions in searching for the best. In the industry, designers were unable to demonstrate or validate the value in their designs through a working prototype as they traditionally did. As a result, they found it difficult to convince leadership to commit to their more innovative designs. They thus often quickly resorted to familiar designs of ML, such as recommenders and reminders (Yang et al. 2018).

How should we sketch and prototype when the design materials in use – large datasets, computational power, time and efforts of data scientists – are costly? Despite efforts to make "ML available to everyone", most designers will face the reality that sufficient data and proficient data scientists are scarce in their teams. A reflective discussion on the current workflow, and how it might adapt to costly ML systems is necessary.

There is also a real need for design tools and methodologies that support designers who lack constant access to capable data scientists. For example, ML tools for designers could simulate the role of the data scientists, enabling designers to quickly evaluate the feasibility of their ideas when sketching. We also see opportunities for research to demonstrate creative designs that use readily available ML solutions (i.e. off-the-shelf ML plugins); designs that do not need intensive ML development effort to implement.

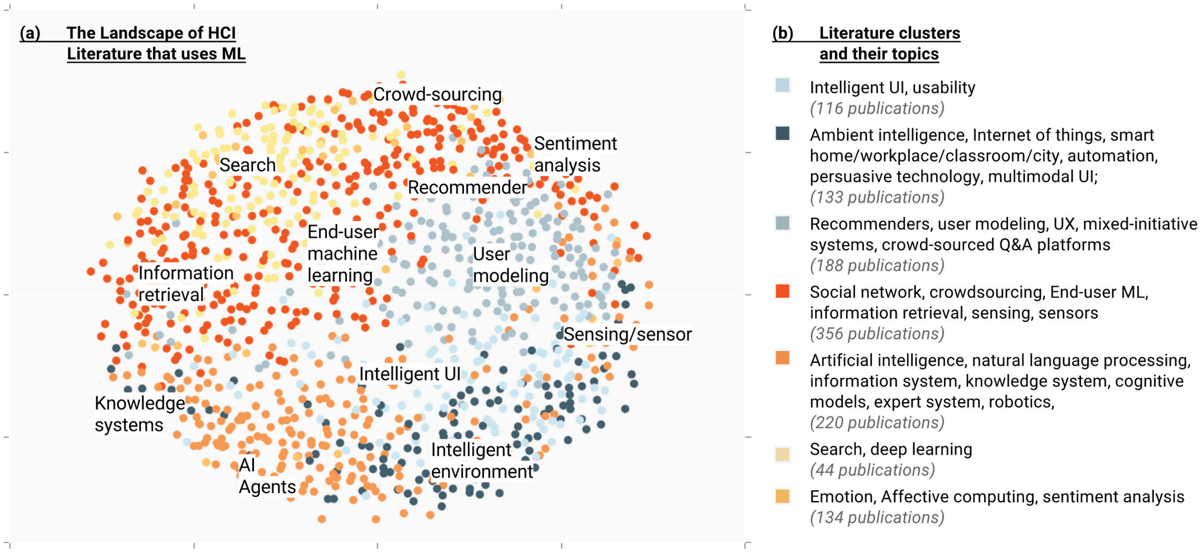


Figure 1: An overview of HCI research that uses machine learning (Yang, Banovic, and Zimmerman 2018). (a) An illustration of the literature landscape based on the semantic distances among each publication. Each dot represents a publication, color-coded by cluster. (b) major topics of each cluster. Topic co-occurrences in these clusters surfaced some common combinations of ML techniques and interaction forms.

UX and ML Match-Making

This leads to the third case in which we worked to help designers identify opportunities in available datasets and algorithms as well as to search for valuable new things to make.

With regard to "technology in searching for users", HCI researchers have previously proposed the *matchmaking* method (Bly and Churchill 1999). Matchmaking starts by asking designers to detail the technical *capabilities* of the tech they are working on. Next, they systematically work to discover *activities* related to these capabilities, *domains* related to the activities, and finally target *users* connected to the revealed domains. Unfortunately, this approach is both under-investigated as a design method and underutilized by both design researchers and practitioners.

> *We wanted to reveal the technical capabilities and interaction forms HCI has worked on with respect to ML. We wanted to identify how ML has been used to generate or augment value for users. Towards these goals, we analyzed 2,494 HCI publications that mention ML with a combination of manual and algorithmic methods. This process produced three representations of ML's design space (Yang, Banovic, and Zimmerman 2018):*
>
> - *7 clusters of work where HCI researchers have repeatedly employed ML techniques to make an advance (Figure 1).*
> - *A schema of machine learning capabilities in terms of enhancing sensing, inference and actuation. Generally, these capabilities increase the value of noisy, real-world data, escalating them to higher-level, more meaningful information.*
> - *4 value channels through which ML advances provide experiential value for users. For example, ML provides inferences about what might be optimal (e.g. when designing MOOC platforms, ML models ideal students' engagement pattern and informs interaction design).*
>
> *The technically defined clusters, capabilities and user value challenges provide starting places to generate unseen designs of ML. Collectively, use of the cluster and model should help design researchers ideate many possible sensitizing concepts. We recommend a process of selecting a technology and then systematically generating ideas from each of the four value channels. This is one form of match-making.*
>
> *Particularly, three clusters of ML technical advances have not yet been bound to particular utilities, interactions or user experiences. These design-wise underutilized clusters are: deep learning, sentiment analysis and social network mining. Designers and researchers can pick one technology material, match it with diverse interaction forms, and generate many possible ways of leveraging it to provide value to users. For example, what can we design when one is able to capture societal happiness on twitter? What values can we provide for users with deep learning, beyond providing more targeted ads and recognizing objects in an image?*

The clusters, schema and value model in this case are neither fixed nor final. They await further examination and discussion. Nonetheless, the exploratory process of categorizing ML's technical capabilities and design characteristics was highly informative to us.

We realized that it seems impossible to categorize ML as

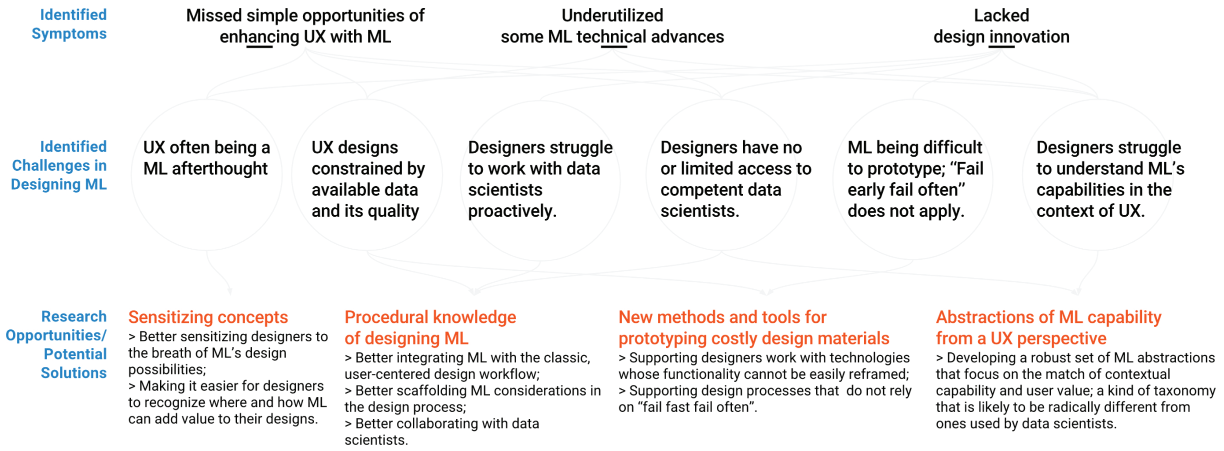

Identified Symptoms
Missed simple opportunities of enhancing UX with ML
Underutilized some ML technical advances
Lacked design innovation
Identified Challenges in Designing ML
UX often being a ML afterthought
UX designs constrained by available data and its quality
Designers struggle to work with data scientists proactively.
Designers have no or limited access to competent data scientists.
ML being difficult to prototype; "Fail early fail often" does not apply.
Designers struggle to understand ML's capabilities in the context of UX.
Research Opportunities/ Potential Solutions
Sensitizing concepts
> Better sensitizing designers to the breath of ML's design possibilities;
> Making it easier for designers to recognize where and how ML can add value to their designs.
Procedural knowledge of designing ML
> Better integrating ML with the classic, user-centered design workflow;
> Better scaffolding ML considerations in the design process;
> Better collaborating with data scientists.
New methods and tools for prototyping costly design materials
> Supporting designers work with technologies whose functionality cannot be easily reframed;
> Supporting design processes that do not rely on "fail fast fail often".
Abstractions of ML capability from a UX perspective
> Developing a robust set of ML abstractions that focus on the match of contextual capability and user value; a kind of taxonomy that is likely to be radically different from ones used by data scientists.

Figure 2: An overview of the challenges in designing ML and the lessons learned in the case studies. These are likely to generalize across many types of ML applications and domains; There is value in investigating these overarching challenges and addressing them systematically.

one design material, and to articulate the capabilities it has or the user activities it can facilitate. For traditional technology materials, for example Bluetooth, researchers provided comprehensive descriptions of its special properties, and then progressed to identify many domains and users these properties might support (Wiberg 2014). In contrast, the capabilities of ML are wedded to its dataset, labels, and underlying algorithm. Its experiential value arises from users' holistic experience over a larger course of interactions. ML in the context of UX design resists easy assimilation into a complete or fixed taxonomy of descriptive mechanisms, such as supervised or unsupervised learning.

Designers need a new way of grasping ML's capabilities, a kind of abstraction that focuses on the match of contextual capability and user value; a kind of taxonomy that is likely to be radically different from ones used by data scientists. The 7 clusters and schema described in the above case provide a starting point for this effort (Figure 1). Additionally, some designers have over the years developed their original abstractions of ML, for example, ML enables *an experience personalized for everyone*", *an evolving relationship with the users*", and "*handling more abstract user instructions*" (Yang et al. 2018). Synthesizing and developing a robust set of such designerly abstractions would help to evolve the understanding of ML as a design material, and to more effectively explore its design possibilities.

Synthesizing the Challenges

The above case studies have illustrated many challenges in innovating ML as a design material as well as the lessons we learned along the way. On reflection, these challenges are not unique to a particular ML problem or application domain, but lie inherently in the characteristics of this design material.

My goal here is not to enumerate the overarching challenges in designing ML, but to highlight the opportunities in addressing them systematically. This section is intended to provide another start point for this emerging area of inquiry (Figure 2).

Inserting ML to Design Practice and Workflow The tension between the classic UCD practice and the designs that arise from available ML systems is a major theme throughout the case studies.

- How to scaffold the design process such that necessary ML considerations can be taken into account timely?
- How to evaluate the technical feasibility of a UX design, especially when proficient data scientists are not constantly accessible?

Without taking ML into consideration throughout product development, even simple designs such as UI adaption could fail. The case studies in this paper demonstrated three kind of workflows; each entailed distinct challenges. Future work should reflect on and improve the workflows we experimented, in searching for better ways of inserting ML to product and service design practice.

At a higher level, the procedural knowledge of designing ML marks a clear space for UX/ML research. Existing work has offered valuable declarative knowledge and conceptual understandings of ML from a design perspective (i.e. algorithm trust, intelligibility, embodiment). Embedding this growing body of new knowledge into organizational and procedural contexts opens up new research opportunities and promises real impact on UX practice.

Sensitizing Designers to Existing ML Design Opportunities Sensitizing UX practitioners to the design opportunities in new technologies is a recurring theme in some HCI research. In previous cases, such as haptics, researchers demonstrate working examplars of the technology to communicate the design possibilities. Interestingly, demonstrating a functional ML system to designers was not enough to sensitize them, as many of these systems have weak connections to divergent user experiences after repeated use.

- How to recognize and draw inspirations from existing design examplars, given that successful ML interactions are often invisible or unremarkable?

- How to recognize ML opportunities with in our own designs, especially when telemetry data are not readily available?

There is a real need to create sensitizing concepts (Zimmerman, Stolterman, and Forlizzi 2010) that communicate ML design opportunities that are not instantly recognizable. Some work is needed to explore the diverse forms of research artifacts and knowledge representations, and to deliberately choose the ones that most effectively sensitize practitioners.

Developing a Designerly Understanding of ML In order to push the boundaries of what ML might be and might do, we need to first bring clarity to its existing design space and to identify major unknown topics as a basis for future research endeavor. The review of HCI/ML literature is a first step towards this goal (Figure 1).

- How to abstract ML's capabilities from a UX perspective?
- How to sketch and evaluate multiple designs when "fail fast, fail often" is practically impossible in prototyping ML systems?

In addition, the case studies suggested an alternative view to the common assumption that teaching designers how ML works as the most effective way of helping them engage with it as a design material. In match-making ML capabilities and UX possibilities, designers comprehend ML in notably different ways than its textbook definitions. Design researchers hoping to aid practitioners might focus on providing designerly abstractions, exemplars, and new tools that help designers grasp the ML's design potential, quickly sketch and prototype, as well as better collaborate with data scientists. We strongly encourage the UX and HCI research community to join us and start a reflective discussion around innovating ML as a design material.

References

Allen, P. v., and Hooker, B. 2017. Useless artificial intelligence, lecture note in internet of enlightened things.

Bly, S., and Churchill, E. F. 1999. Design through matchmaking: technology in search of users. *interactions* 6(2):23–31.

Council, D. 2005. The 'double diamond' design process model. *Design Council*.

Dove, G.; Halskov, K.; Forlizzi, J.; and Zimmerman, J. 2017. UX Design Innovation: Challenges for Working with Machine Learning as a Design Material. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems - CHI '17*, 278–288. New York, New York, USA: ACM Press.

Gillies, M.; Lee, B.; D'Alessandro, N.; Tilmanne, J.; Kulesza, T.; Caramiaux, B.; Fiebrink, R.; Tanaka, A.; Garcia, J.; Bevilacqua, F.; Heloir, A.; Nunnari, F.; Mackay, W.; and Amershi, S. 2016. Human-Centred Machine Learning. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems - CHI EA '16*, 3558–3565. New York, New York, USA: ACM Press.

Hebron, P. 2016. *Machine learning for designers*. O'Reilly Media.

Kuniavsky, M.; Churchill, E.; and Steenson, M. W. 2017. The 2017 aaai spring symposium series technical reports: Designing the user experience of machine learning systems. Technical Report SS-17-04, Palo Alto, California.

Louridas, P. 1999. Design as bricolage: anthropology meets design thinking. *Design Studies* 20(6):517–535.

Moussette, C. 2012. *Simple haptics: Sketching perspectives for the design of haptic interactions*. Ph.D. Dissertation, Umeå Universitet.

Schön, D., and Bennett, J. 1996. Reflective conversation with materials. In *Bringing design to software*, 171–189. ACM.

Schön, D. A. 1984. *The reflective practitioner: How professionals think in action*, volume 5126. Basic books.

Wiberg, M. 2014. Methodology for materiality: interaction design research through a material lens. *Personal and ubiquitous computing* 18(3):625–636.

Yang, Q.; Banovic, N.; and Zimmerman, J. 2018. Mapping Machine Learning Advances from HCI Research to Reveal Starting Places for Design Research. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems - CHI '18*.

Yang, Q.; Zimmerman, J.; Steinfeld, A.; Carey, L.; and Antaki, J. F. 2016a. Investigating the heart pump implant decision process: Opportunities for decision support tools to help. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, 4477–4488. New York, NY, USA: ACM.

Yang, Q.; Zimmerman, J.; Steinfeld, A.; and Tomasic, A. 2016b. Planning Adaptive Mobile Experiences When Wireframing. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems - DIS '16*, 565–576. Brisbane, QLD, Australia: ACM Press.

Yang, Q.; Sciuto, A.; Steinfeld, A.; Forlizzi, J.; and Zimmerman, J. 2018. Investigating how experienced UX designers effectively work with machine learning. Manuscript submitted for publication.

Zimmerman, J.; Stolterman, E.; and Forlizzi, J. 2010. An analysis and critique of research through design: towards a formalization of a research approach. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems*, 310–319. ACM.

Integrating Representation, Reasoning, Learning, and Execution for Goal Directed Autonomy

Robot Behavioral Exploration and Multi-Modal Perception Using Dynamically Constructed Controllers

Saeid Amiri,[1] Suhua Wei,[1] Shiqi Zhang,[1] Jivko Sinapov,[2] Jesse Thomason,[3] Peter Stone[3]
[1] Department of Electrical Engineering and Computer Science, Cleveland State University
[2] Department of Computer Science, Tufts University
[3] Department of Computer Science, The University of Texas at Austin
{s.amiri@vikes; s.wei@vikes; s.zhang9@}.csuohio.edu; jsinapov@cs.tufts.edu
t{jesse; pstone}@cs.utexas.edu

Abstract

Intelligent robots frequently need to explore the objects in their working environments. Modern sensors have enabled robots to learn object properties via perception of multiple modalities. However, object exploration in the real world poses a challenging trade-off between information gains and exploration action costs. Mixed observability Markov decision process (MOMDP) is a framework for planning under uncertainty, while accounting for both fully and partially observable components of the state. Robot perception frequently has to face such mixed observability. This work enables a robot equipped with an arm to dynamically construct query-oriented MOMDPs for object exploration. The robot's behavioral policy is learned from two datasets collected using real robots. Our approach enables a robot to explore object properties in a way that is significantly faster while improving accuracies in comparison to existing methods that rely on hand-coded exploration strategies.

1 Introduction

Service robots are increasingly present in everyday environments, such as homes, offices, airports, and hospitals, where a common task is to retrieve an object for a user. Consider the request, "*Please fetch me the red, empty bottle*." A key problem for the robot is to decide whether a particular candidate object matches the properties in the query. For certain words (e.g., *heavy*, *soft*, etc.), visual classification of the object is insufficient as the robot would need to perform an action (e.g., lift the object to determine whether it is heavy or not). Multi-modal perception research has focused on combining information arising from such multiple sensory modalities.

Given multi-modal perception capabilities, a robot needs to decide which actions (possibly out of many) to perform on an object, i.e., generate a behavioral policy for a given request. For instance, to obtain an object's color, a robot simply needs to adjust the pose of its camera, whereas sensing the content of a container requires two actions: grasping and shaking. The robot needs to select actions in such a way that the information gain about object properties is maximized while the cost of actions is minimized. It should be noted that the robot needs to use sequential reasoning in this action selection process, e.g., a shaking action would make sense only if a grasping action has been (successfully) executed. Also, robot perception capabilities are imperfect, so the robot sometimes needs to take the same action more than once. Probabilistic planning algorithms aim at computing action policies to help select actions toward maximizing long-term utility (information gain in our case), while considering the uncertainty in non-deterministic action outcomes.

Markov decision processes (MDPs) (Puterman 1994) and partially observable MDPs (POMDPs) (Kaelbling, Littman, and Cassandra 1998) enable an agent to plan under uncertainty with full and partial observability respectively. However, the observability of real-world domains is frequently mixed: some components of the current state can be fully observable while others are not. A mixed observability Markov decision process (MOMDP) is a special form of POMDP that accounts for both fully and partially observable components of the state (Ong et al. 2010). In this work, we model robot multi-modal perception problems using MOMDPs because of the mixed observability of the world that the robot interacts with (e.g., whether an object is in hand or not is fully observable, but object properties such as color and weight are not). Referring to our model as a MOMDP (as opposed to a POMDP) is not of practical importance in this paper. It is mainly for ease of describing the domain.

Robot behavioral exploration policies are learned from the experience of a robot interacting with objects in the real world. We use datasets that include tens of objects and nearly one hundred properties. In such domains, it frequently takes a prohibitively long time to compute effective behavioral exploration policies. To tackle this issue, we dynamically construct MOMDP-based controllers to model a minimum set of domain variables that are relevant to current user queries (e.g. "red, empty bottle"). This strategy ensures a small state set and enables us to generate high-quality robot action policies in a reasonable time (e.g., ≤ 2 seconds). Our experiments show that the policies of the constructed controllers improve recognition accuracy and reduce exploration cost when compared to baseline strategies that deterministically or randomly use predefined sequences of actions.

2 Related Work

Recent research in robotics has shown that robots can learn to classify objects using computer vision methods as well as non-visual perception coupled with actions performed on the objects (Högman, Björkman, and Kragic 2013; Sinapov et al. 2014; Thomason et al. 2016). For example, a robot can learn to determine whether a container is full or not based on the sounds produced when shaking the container (Sinapov and Stoytchev 2009); or learn whether an object is soft or hard based on the haptic sensations produced when pressing it (Chu et al. 2015). Past work has shown that robots can associate (or *ground*) these sensory perceptions with human language predicates in vision space (Alomari et al. 2017; Whitney et al. 2016; Krishnamurthy and Kollar 2013; Matuszek et al. 2012) and joint visual and haptic spaces (Gao et al. 2016).

Nevertheless, there has been relatively little emphasis on enabling a robot to *efficiently* select actions at test time when it is tasked with classifying a new object. The few approaches for tackling action selection, e.g., (Rebguns, Ford, and Fasel 2011; Fishel and Loeb 2012; Sinapov et al. 2014), assume that only one target property needs to be identified (e.g., the object's identity in the case of object recognition). In contrast, we address the problem where a robot needs to recognize multiple properties about an object, e.g., "is the object a *red empty bottle*?".

Sequential decision-making frameworks, such as MDPs, POMDPs and MOMDPs, can be used for probabilistic planning toward achieving long-term goals, while accounting for non-deterministic action outcomes and different observabilities (Kaelbling, Littman, and Cassandra 1998; Ong et al. 2010). As a result, these frameworks have been applied to object exploration in robotics. For instance, POMDPs were used for suggesting visual operators and regions of interests for exploring multiple objects on a tabletop scenario (Sridharan, Wyatt, and Dearden 2010), and more recent work used a robotic arm to move objects enabling better visual analysis (Pajarinen and Kyrki 2015). However, interaction with objects in these lines of research relies heavily on robot vision while other sensing modalities, such as audio and haptics, are not considered.

Behavioral policies of multi-modal object exploration have been learned in simulation using deep reinforcement learning methods (Denil et al. 2017), where *force* was directly used in the interactions with objects. The simulation environment used in that work makes it possible to run large numbers of trials, but limits its applicability on real robots.

3 Theoretical Framework

Next, we describe the theoretical framework used by the robot to learn predicate recognition models and generate efficient policies when tasked with identifying whether a set of predicates hold true for a new object.

3.1 Multi-Modal Predicate Learning

In this work, the robot learns predicate recognition models using the methodology described in (Sinapov, Schenck, and Stoytchev 2014; Thomason et al. 2016), briefly summarized here. In this methodology, the robot uses behaviors (e.g., *look*, *grasp*, *lift*) coupled with sensory modalities (e.g., *color*, *haptics*, *audio*) to identify whether a predicate (i.e., a word that a human may use to describe an object) holds true for an object.

Let $\mathcal{P}$ be the set of predicates, let $\mathcal{B}$ be the set of behaviors (i.e., actions), and let $\mathcal{C}$ be the set of sensorimotor contexts, where each context $c \in \mathcal{C}$ corresponds to a combination of a behavior and sensory modality (e.g., *look-color*, *lift-haptics*). For each predicate p, and context c, the robot learns a classifier using data points $[x_i^c, y_i]$, where x_i^c is the i^{th} observation feature vector in context c, and $y_i = true$ if the predicate p holds true for the object in trial i, and $false$ otherwise.

Let $\mathcal{C}_b \subset \mathcal{C}$ be the set of sensorimotor contexts associated with behavior $b \in \mathcal{B}$. When executing action b, the robot queries the classifiers associated with contexts $\mathcal{C}_b$ and combines their outputs to estimate a score (normalized in the range of 0.0 to 1.0) for each predicate $p \in \mathcal{P}$. In other words, each behavior acts as a classifier itself. At the end of the training stage, the robot performs internal cross-validation and stores the confusion matrix $C_p^b \in \mathbb{R}^{2\times 2}$ for predicate p and behavior b. Next, we describe the problem of generating an action policy when identifying whether a set of predicates hold true for an object that was not present during training.

3.2 MOMDP-based Controllers

Behaviors (or actions[1]), such as *look* and *drop*, have different costs and different accuracies in predicate recognition. At each step, the robot has to decide whether more exploration behaviors are needed, and, if so, select the exploration behavior that produces the most information. In order to sequence these behaviors toward maximizing information gain, subject to the cost of each behavior (e.g., the time it takes to execute it), it is necessary to further consider preconditions and non-deterministic outcomes of the actions. For instance, *shaking* and *dropping* actions make sense only if a preceding *grasping* action succeeds; and, in practice, *grasping* actions are unreliable and succeed with probability.

In this work, we assume action outcomes are fully observable and object properties are not. For instance, a robot can reliably sense whether a *grasping* action is successful, but it cannot reliably sense the color of a bottle or whether that bottle is full. Due to this mixed observability and unreliable action outcomes, we use mixed observability MDPs (MOMDPs) (Ong et al. 2010) to model the sequential decision-making problem for object exploration. We next present how we formalize our object exploration problem within the MOMDP framework.

A MOMDP is fundamentally a factored POMDP with mixed state variables. The fully observable state components are represented as a single state variable x (in our case, the *robot-object status*, e.g., the object is in hand or not), while the partially observable components are represented as state

[1] The terms of "behavior" and "action" are used interchangeably in this paper.

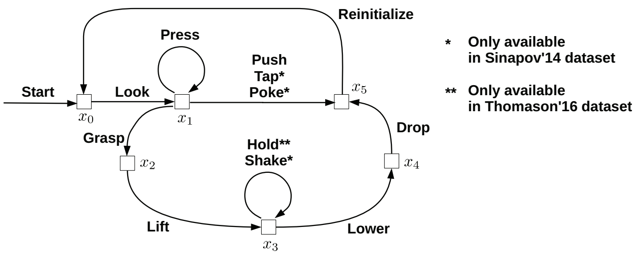
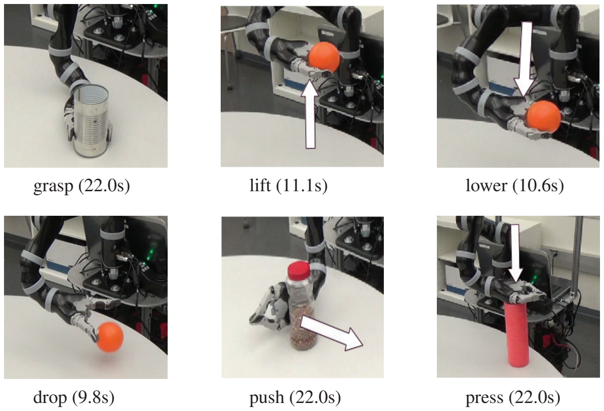


Figure 1: A simplified version of the transition diagram in space $\mathcal{X}$ for object exploration. This figure only shows the probabilistic transitions led by *exploration actions*. *Report actions* that deterministically lead transitions from $x_i \in \mathcal{X}$ to the *term* state are not included.



grasp (22.0s) lift (11.1s) lower (10.6s)

drop (9.8s) push (22.0s) press (22.0s)

Figure 2: The behaviors, and their durations in seconds (behaviors are from the **Thomason16** dataset detailed in Sec. 4). In addition, the *hold* (1.0s) behavior was performed by holding the object in place. The *look* (0.5s) behavior was also performed by taking a visual snapshot of the object using the robot's sensors prior to exploration.

variable y (in our case, the *object properties*, e.g., the object is heavy or not). As a result, (x,y) specifies the complete system state, and the state space is factored as $S = \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is the space for fully observable variables and $\mathcal{Y}$ is the space for partially observable variables.

Formally, a MOMDP model is specified as a tuple,

$$(\mathcal{X},\ \mathcal{Y},\ A,\ T_{\mathcal{X}},\ T_{\mathcal{Y}},\ R,\ Z,\ \mathcal{O},\ \gamma),$$

where A is the action set, $T_{\mathcal{X}}$ and $T_{\mathcal{Y}}$ are the transition functions for fully and partially observable variables respectively, R is the reward function, Z is the observation set, $\mathcal{O}$ is the observation function, and γ is the discount factor.

The definitions of A, R, Z, $\mathcal{O}$, and γ of a MOMDP are identical to these of POMDPs (Kaelbling, Littman, and Cassandra 1998), except that Z and $\mathcal{O}$ are only applicable to $\mathcal{Y}$, the partially observable components of the state space. γ is the discount factor that specifies the planning horizon. We formalize our object exploration problem as a MOMDP (as a special form of POMDP) mainly for ease of describing the fully and partially observable variables in our domain.

Next, we present how each component of our MOMDP model is specified for our object exploration problem.

3.3 State Space Specification

The state space of our MOMDP-based controllers has two components of $\mathcal{X}$ and $\mathcal{Y}$. The global state space S includes a Cartesian product of $\mathcal{X}$ and $\mathcal{Y}$,

$$S = \{(x,y) \mid x \in \mathcal{X} \text{ and } y \in \mathcal{Y}\}$$

$\mathcal{X}$ is the state set specified by fully observable domain variables. In our case, $\mathcal{X}$ includes a set of six states $\{x_0, \cdots, x_5\}$, as shown in Figure 1, and a terminal state $term \in \mathcal{X}$ that identifies the end of an episode. $x \in \mathcal{X}$ is fully observable, and the robot knows the current state of the robot-object system, e.g., whether grasping and dropping actions are successful or not.

$\mathcal{Y}$ is the state set specified by partially observable domain variables. In our case, these variables correspond to N object properties that are queried about, $\{v_0,\ v_1,\ \cdots,\ v_{N-1}\}$, where the value of v_i is either *true* or *false*. Thus, $|\mathcal{Y}| = 2^N$.

For instance, given an object description that includes three properties (e.g., "a *red empty bottle*"), $\mathcal{Y}$ includes $2^3 = 8$ states. Since $y \in \mathcal{Y}$ is partially observable, it needs to be estimated through observations. It should be noted that there is no state transition in the space of $\mathcal{Y}$, as we assume object properties do not change over the course of robot action.

3.4 Actions and Transition System

We present the transition system of our MOMDP-based controllers by first introducing the action set and then the transition probabilities. $A : A^e \cup A^r$ is the action set. A^e includes the object *exploration* actions pulled from the literature of robot exploration, as shown in Figure 1, and A^r includes the *reporting* actions used for object property identification.

Exploration actions: Figure 1 shows all exploration actions except for action *ask* that is allowed in any state $x \in \mathcal{X}$. Among the actions, *tap*, *poke*, and *shake* are only available in the dataset of (Sinapov, Schenck, and Stoytchev 2014) and *hold* is only available in the dataset of (Thomason et al. 2016). As one of the main contributions, our approach enables a robot to automatically figure out what actions are useful given a user query by learning from the datasets. Pictures of a robot executing some of the exploration actions are shown in Figure 2.

Reporting actions: A^r includes a set of actions that are used for reporting the object's properties and can deterministically lead the state transition to *term* (terminal state). For instance, if a user queries about "a blue, heavy can", there will be three binary variables specifying each of properties is true or false. As a result, there will be eight reporting actions. For $a \in A^r$, we use $s \odot a$ (or $y \odot a$) to represent that the report of a matches the underlying values of object properties (i.e., a correct report) and use $s \oslash a$ (or $y \oslash a$) otherwise.

$T_{\mathcal{X}} : \mathcal{X} \times A \times \mathcal{X} \to [0,1]$ is the state transition function in the fully observable component of the current state. $T_{\mathcal{X}}$ includes a set of conditional probabilities of transitions from $x \in \mathcal{X}$—the fully observable component of the current state—to $x' \in \mathcal{X}$, the component of the next state, given $a \in A$ the current action. Reporting actions and illegal exploration actions (e.g., *dropping* an object in state x_1—before a successful grasp) lead state transitions to *term* with 1.0 probability.

Most exploration actions are unreliable and succeed probabilistically. For instance, $p(x_4,\ drop,\ x_5) = 0.95$ in our case, indicating there is small probability the object is stuck in the robot's hand. The success rate of action *look* is 1.0 in our case, since without changing positions of either the camera or the object it does not make sense to keep running the same vision algorithms and hence it is not allowed.

$T_{\mathcal{Y}} : \mathcal{Y} \times A \times \mathcal{Y} \to [0,1]$ is the state transition function in the partially observable component of the current state. It is an identity matrix in our case, (we assume) because object properties do not change during the process of the robot's exploration actions.

3.5 Reward Function and Discount Factor

$R : S \times A \to \mathbb{R}$ is the reward function. Each *exploration action*, $a^e \in A^e$, has a cost that is determined by the time required to complete the action. These costs are empirically assigned according to the datasets used in this research. The costs of *reporting actions* depend on whether the report is correct.

$$R(s,a) = \begin{cases} r^-, & \textbf{if } s \in S,\ a \in A^r,\ s \oslash a \\ r^+, & \textbf{if } s \in S,\ a \in A^r,\ s \odot a \end{cases}$$

where r^- (or r^+) is negative (or positive) given an incorrect (or correct) report. Unless otherwise specified, $r^- = -500$ and $r^+ = 500$ in this paper.

Costs of other exploration actions are within the range of $[0.5, 22.0]$ (corresponding reward is negative), except that action *ask* has the cost of 100.0. γ is a discount factor, and $\gamma = 0.99$ in our case. This setting gives the robot a relatively long planning horizon.

3.6 Observations and Observation Function

$Z : Z^h \cup \emptyset$ is a set of observations. Elements in Z^h include all possible combinations of object properties and have one-one correspondence to elements in A^r and $\mathcal{Y}$. For instance, when the query is about "a *red empty bottle*", there exists an observation $z \in Z^h$ that represents "the object's color is red; it is not empty, and it is a bottle." Actions that produce no information gain (*reinitialize*, in our case), and reporting actions in A^r result in a $\emptyset$ (none) observation.

$O : S \times A \times Z \to [0,1]$ is the observation function that specifies the probability of observing $z \in Z$ when action a is executed in state s: $O(s,a,z)$. In this work, the probabilities are learned from performing cross-validation on the robot's training data. As described in Section 3.1, predicate learning produces confusion matrix $C^b_p \in \mathbb{R}^{2\times 2}$ for each predicate p and each behavior b, where b corresponds to one of the exploration actions shown in Figure 1.

$$\begin{aligned} O(s,a,z) &= Pr(\mathbf{p}^s, b, \mathbf{p}^z) \\ &= C^b_{p_0}(p^s_0, p^z_0) \cdot C^b_{p_1}(p^s_1, p^z_1) \cdots C^b_{p_{N-1}}(p^s_{N-1}, p^z_{N-1}) \end{aligned}$$

where behavior b corresponds to action a; $\mathbf{p}^s$ and $\mathbf{p}^z$ are the vectors of *true* and *observed* values (0 or 1) of the predicates; p^s_i (or p^z_i) is the true (or observed) value of the i^{th} predicate; and N is the total number of predicates in the query.

3.7 Dynamically Constructed Controllers

State set $\mathcal{Y}$ can be very large, due to the large number of predicates and the exponentially increasing number of their combinations. For example, one of the datasets in our experiments contains 81 predicates, resulting in 2^{81} possible states. Due to limited computational resources, it would be intractable for a robot to generate a far-sighted policy for identifying an object according to all 81 predicates.

Recent research decomposes a sequential decision-making problem into two tractable subproblems that respectively focus on high-dimensional reasoning (e.g., objects with many properties) and long-horizon planning (e.g., tasks that require many actions) (Zhang, Khandelwal, and Stone 2017). Based on that approach, we dynamically construct controllers that include a minimum set of predicates, instead of modeling all of them, in the $\mathcal{Y}$ component. In addition to $\mathcal{Y}$, the following components depend on the user query: reporting actions A^r, object property combinations Z^h, and the reward and observation functions (due to the involvement of $\mathcal{Y}$). As a result, our query-oriented, MOMDP-based controllers are relatively very small, and typically include fewer than 100 states at runtime.

It should be noted that we use MOMDP, as a special form of POMDP, to model our domain mainly for the ease of describing the mixed observability over $\mathcal{X}$ and $\mathcal{Y}$ (Section 3.3). Our approach enables automatic generation of complete MOMDP models. One can encode such MOMDP models in such a way that existing POMDP solvers (e.g., (Kurniawati, Hsu, and Lee 2009)) can be used to generate policies, as we do in this work.

4 Experimental Results

We evaluate the proposed method using two datasets in which a robot explored a set of objects using a variety of exploratory behaviors and sensory modalities, and show that for both our proposed MOMDP model outperforms baseline models in exploration accuracy and overall exploration cost. Two datasets of **Sinapov14** and **Thomason16** have been used in the experiments, where **Thomason16** has a much more diverse set of household objects and a larger number of predicates that arose naturally during human-robot interaction gameplay.

Sinapov14 Dataset: In this dataset, the robot explored 36 different objects using 11 prototypical exploratory behaviors: *look*, *grasp*, *lift*, *shake*, *shake-fast*, *lower*, *drop*, *push*, *poke*, *tap*, and *press* 10 different times per object. The objects are lidded containers with the same shape and varied

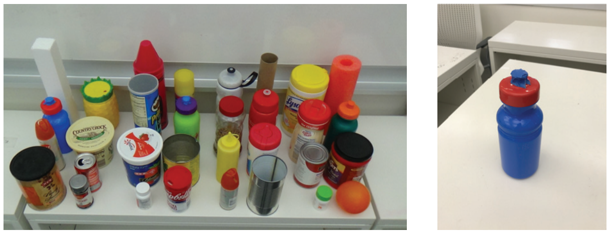


Figure 3: Objects in the **Thomason16** dataset (Left) and the one used in the illustrative example in Section 4.1 (Right).

along 3 different attributes: 1) color: *red*, *green*, *blue*; 2) weight: *light*, *medium*, *heavy*; and 3) contents: *beans*, *rice*, *glass*, *screws*. These variations result in the $3 \times 3 \times 4 = 36$ objects bearing combinations of these attributes in the set P that the robot is tasked with learning. It should be noted that costs of actions in the two datasets are different, because the datasets were collected using different robots.

Thomason16 Dataset: In this dataset, the robot explored 32 common household objects using 8 exploratory actions: *look*, *grasp*, *lift*, *hold*, *lower*, *drop*, *push*, and *press*. Each behavior was performed 5 times on each object. The dataset was originally produced for the task of learning how sets of objects can be ordered and is described in greater detail by (Sinapov et al. 2016).

For the *look* behavior, *color*, *shape*, and *deep* features (the penultimate layer of the trained VGG network (Simonyan and Zisserman 2014)) are available. For the remaining behaviors, the robot recorded *audio*, *proprioceptive* (finger positions for *grasp*), and *haptic* (i.e., joint forces) features produced by the interaction with the object. These modalities result in $|C| = 7 \times 2 + 1 \times 3 = 17$ sensorimotor contexts.

The set of predicates $\mathcal{P}$ consisted of 81 words used by human participants to describe objects in this dataset during an interactive gameplay scenario described by (Thomason et al. 2016). Example predicates include the words *red*, *heavy*, *empty*, *full*, *cylindrical*, *round*, etc. Unlike the **Sinapov14** dataset, here the objects vary greatly, and the predicate recognition problem is much more difficult.

4.1 Illustrative Example

We now describe an example in which a robot is tasked with identifying properties of a given object. We randomly selected an object from the **Thomason16** dataset: a blue and red bottle full of water (Figure 3). We then randomly selected properties, in this case "yellow" and "metallic," and asked the robot to identify whether the object has each of the properties or not. The selected object was not part of the robot's training set used to learn the predicate recognition models and the MOMDP observation model. The robot should report negative to both properties while minimizing the overall cost of exploration actions.

Given this user query, we generate a MOMDP model that includes 25 states. We then generate an action policy using past work's methods (Kurniawati, Hsu, and Lee 2009).

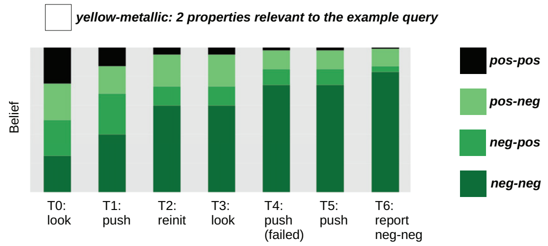


Figure 4: Action selection and belief change in the exploration of a red and blue bottle full of water, given a query of *yellow* and *metallic*.

Currently, building the model takes almost no time, and we uniformly gave two seconds for policy generation using the model (same in all experiments). The time for computing the policy is insignificant relative to the time for exploratory behaviors (which is what we are really trying to minimize).

Figure 4 shows the belief change in this process. The initial distributions over $\mathcal{X}$ and $\mathcal{Y}$ are $[1.0, 0.0, \cdots]$ and $[0.25, 0.25, 0.25, 0.25]$ respectively. The policy suggests "look" first. We queried the dataset to make an observation, *neg-neg* in this case. The belief over $\mathcal{Y}$ is updated based on this observation: $[0.41, 0.28, 0.19, 0.13]$, where the entries represent *neg-neg*, *neg-pos*, *pos-neg*, and *pos-pos* respectively. There is a (fully observable) state transition in $\mathcal{X}$, from x_0 to x_1, so the belief over $\mathcal{X}$ becomes $[0.0, 1.0, 0.0, \cdots]$. Based on the updated beliefs, the policy suggests taking the "push" action, which results in another *neg-neg* observation. Accordingly, the belief over $\mathcal{Y}$ is updated to $[0.60, 0.13, 0.22, 0.05]$, which indicates that the robot is more confident that the object is neither "yellow" nor "metallic". After actions of *reinitialize*, *look*, *push*, and *push* (this first *push* action was unsuccessful, and produced the $\emptyset$ observation), the belief over $\mathcal{Y}$ becomes $[0.84, 0.04, 0.12, 0.01]$. The policy finally suggests reporting *neg-neg*, making it a successful trial with an overall cost of 167 seconds, which results in a reward of $500 - 167 = 333$ (an incorrect report would have resulted in -667 reward).

Remarks: It should be noted that the classifiers associated with each behavior and word will produce an output even in cases where the sensory signals from that behavior are irrelevant to the word. For instance, although the sensory signals relevant to "push" are haptics and audio, the first "push" action results in an observation of "yellow". It was "yellow:neg", because the training set prior of most objects are not yellow. The robot favors actions that distinguish 'easy' predicates (*look* distinguishes *yellow* well in this case) because there is the discount factor (0.99): If an action is useful, the robot will prefer taking it early. The more the action is delayed, the more the expected reward is discounted.

4.2 Results

Next, we describe the experiments we conducted to evaluate the proposed MOMDP-based multi-modal perception strategy for object exploration. The goal was to increase the

Table 1: Performances of MOMDP-based and two baseline planners in cost (second) and accuracy on the **Sinapov14** dataset. Numbers in parenthesis denote the Standard Deviations over 400 trials.

Properties	Method	Overall cost (std)	Accuracy
Two	Random Plus	17.56 (30)	0.245
	Predefined Plus	37.10 (0.00)	0.583
	MOMDP (Ours)	29.85 (12.87)	0.860
Three	Random Plus	10.12 (21.77)	0.130
	Predefined Plus	37.10 (0.00)	0.373
	MOMDP (Ours)	33.87 (8.78)	0.903

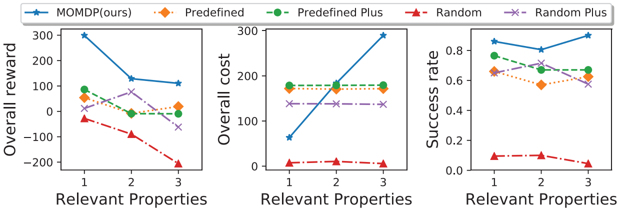


Figure 5: Evaluations of five actions strategies on the **Thomason16** dataset. Comparisons are made in three categories of *overall reward* (Left), *exploration cost* (Middle), and *success rate* (Right).

accuracy in identifying properties of a novel object while reducing the overall action costs required in this process. In all evaluation runs, the object that needs to be identified was not part of the robot's training set when learning the predicate recognition models or the MOMDP parameters. The following baseline action strategies are used in experiments, where belief is updated using Bayes' rule except for *Random*:

- *Random*: Actions are randomly selected from A that includes both reporting and legal exploration actions. A trial is terminated any of the reporting actions.
- *Random Plus*: Actions are randomly selected from legal exploration actions. Under an exploration budget, one selects the reporting action that makes the best sense (i.e., that corresponding to y with the highest belief).
- *Predefined*: An action sequence is strictly followed: *ask, look, press, grasp, lift, lower* and *drop*.[2] Under an exploration budget or in early terminations caused by illegal actions, the robot selects the reporting action that makes the best sense.
- *Predefined Plus*: The same as *Predefined* except that unsuccessful actions are repeated until achieving the desired result(s).

Sinapov14 Dataset: In each trial, we place an object that has three attributes (color, weight and content) on a table and then generate an object description that includes the values of two or three attributes. This description matches the object in only half of the trials. When two (or three) attributes are queried, $\mathcal{Y}$ includes four (or eight) states plus *term* state, resulting in $\mathcal{S}$ that includes 25 (or 49) states. The other components of the dynamically constructed MOMDPs grow accordingly, given an increasing number of queried attributes.

Experimental results are reported in Table 1. Not surprisingly, randomly selecting actions produces low accuracy. The overall cost is smaller in more challenging trials (all three properties are questioned), because in these trials there are relatively fewer exploration actions (more properties produce more reporting actions), making the agent more likely to take a reporting action. Our MOMDP-based multimodal perception strategy reduces the overall action cost while significantly improving the reporting accuracy. Our performance improvement is achieved by repeating actions as needed, selecting legal actions (e.g., *lift* is legal only if the current state is x_2) that produce the most information or have the potential of doing so in the future, and even arbitrarily reporting without "wasting" exploration actions given queries where the exploration actions are not effective.

Thomason16 Dataset: In this set of experiments, a user query is specified by randomly selecting one object and N properties ($1 \leq N \leq 3$), on which the robot is questioned. Each data point is an average over 200 trials, where we conducted pairwise comparisons over the five strategies, i.e., the strategies were evaluated using the same set of user queries. A trial is successful only if the robot reports correctly on all properties. It should be noted that most of the contexts are misleading in this dataset due to the large number of object properties, so it happens that more exploration actions confuse the robot more if the actions are not carefully selected. Figure 5 shows the experimental results. Overall reward is computed by subtracting overall action cost from the reward yielded by the reporting action (either a big bonus or a big penalty). We do not compute standard deviations in this dataset, because the diversity of the tasks results in problems of very different difficulties.

We can see our MOMDP-based strategy consistently performs the best in terms of the overall reward and overall accuracy. When more properties are queried, the MOMDP-based controllers enable the robot to take more exploration actions (Middle subfigure), whereas the baselines could not adjust their question-asking strategy accordingly.

The last experiment aims to experimentally evaluate the need of dynamically constructed controllers. We constructed MOMDP controllers including two relevant and an increasing number of irrelevant properties (i.e., the ones that are not queried). Results are shown in Figure 6. We can see, the quality of the generated action policies decreases soon (from higher than 150 to lower than 25 in reward), when more irrelevant properties are included in the MOMDPs. We did not include six or more irrelevant properties, because the solver cannot produce any policy in one and a half minutes.

[2] Action *ask* was used only in the **Thomason16** experiments, because other exploration actions are not as effective as in **Sinapov14**.

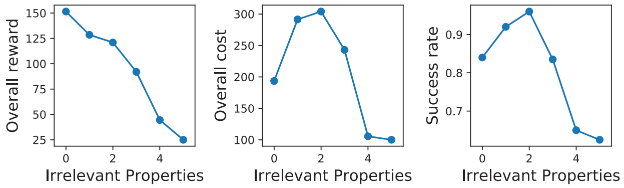


Figure 6: A “super” MOMDP that models two relevant and (an increasing number of) irrelevant properties, in comparison to dynamically constructed controllers used in this work.

5 Conclusions and Future Work

We investigate using mixed observability Markov decision processes (MOMDPs) to help robots select actions for multimodal perception in object exploration tasks. Our approach can dynamically construct a MOMDP model given an object description from a human user (e.g., “*a blue heavy bottle*”), compute a high-quality policy for this model, and use the policy to guide robot behaviors (such as “look” and “shake”) toward maximizing information gain. The dynamically built controllers enable the robot to focus on a minimum set of domain variables that are relevant to the current object and query. The MOMDP models are constructed using two existing datasets collected with robots interacting with objects in the real world. Experimental results show that our object exploration approach enables the robot to identify object properties more accurately without introducing extra cost from exploration actions compared to a baseline that suggests actions following a predefined action sequence.

This research primarily focuses on a robot exploring objects in a tabletop scenario. In future work, we plan to investigate applying this approach to tasks that involve more human-robot interaction and mobile robot platforms, where exploration would require navigation actions and perceptual modalities such as human-robot dialog. Finally, in the two datasets used in this paper, the robot’s manipulation actions were always successful but that would not always be the case in a real-world scenario; therefore we plan to extend our framework to situations in which the robot’s actions may fail (in terms of manipulation) or cause undesirable outcomes (e.g., dropping an object may break it).

Acknowledgments

A portion of this work has taken place in the Learning Agents Research Group (LARG) at UT Austin. LARG research is supported in part by NSF (IIS-1637736, IIS-1651089, IIS-1724157), Intel, Raytheon, and Lockheed Martin. Peter Stone serves on the Board of Directors of Cogitai, Inc. The terms of this arrangement have been reviewed and approved by the University of Texas at Austin in accordance with its policy on objectivity in research.

References

Alomari, M.; Duckworth, P.; Hogg, D. C.; and Cohn, A. G. 2017. Natural language acquisition and grounding for embodied robotic systems. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 4349–4356.

Chu, V.; McMahon, I.; Riano, L.; McDonald, C. G.; He, Q.; Perez-Tejada, J. M.; Arrigo, M.; Darrell, T.; and Kuchenbecker, K. J. 2015. Robotic learning of haptic adjectives through physical interaction. *Robotics and Autonomous Systems* 63:279–292.

Denil, M.; Agrawal, P.; Kulkarni, T. D.; Erez, T.; Battaglia, P.; and de Freitas, N. 2017. Learning to perform physics experiments via deep reinforcement learning. In *International Conference on Learning Representations*.

Fishel, J., and Loeb, G. 2012. Bayesian exploration for intelligent identification of textures. *Frontiers in Neurorobotics* 6:4.

Gao, Y.; Hendricks, L. A.; Kuchenbecker, K. J.; and Darrell, T. 2016. Deep learning for tactile understanding from visual and haptic data. In *International Conference on Robotics and Automation*, 536–543. IEEE.

Högman, V.; Björkman, M.; and Kragic, D. 2013. Interactive object classification using sensorimotor contingencies. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2799–2805. IEEE.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial Intelligence* 101(1):99–134.

Krishnamurthy, J., and Kollar, T. 2013. Jointly learning to parse and perceive: Connecting natural language to the physical world. *Transactions of the Association for Computational Linguistics* 1:193–206.

Kurniawati, H.; Hsu, D.; and Lee, W. S. 2009. SARSOP: efficient point-based POMDP planning by approximating optimally reachable belief spaces. In *Robotics: Science and Systems Conference*, 65–72. The MIT Press.

Matuszek, C.; FitzGerald, N.; Zettlemoyer, L.; Bo, L.; and Fox, D. 2012. A joint model of language and perception for grounded attribute learning. In *Proceedings of the 29th International Conference on Machine Learning*.

Ong, S. C.; Png, S. W.; Hsu, D.; and Lee, W. S. 2010. Planning under uncertainty for robotic tasks with mixed observability. *The International Journal of Robotics Research* 29(8):1053–1068.

Pajarinen, J., and Kyrki, V. 2015. Robotic manipulation of multiple objects as a POMDP. *Artificial Intelligence*.

Puterman, M. L. 1994. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.

Rebguns, A.; Ford, D.; and Fasel, I. R. 2011. Infomax control for acoustic exploration of objects by a mobile robot. In *Lifelong Learning*.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

Sinapov, J., and Stoytchev, A. 2009. From acoustic object recognition to object categorization by a humanoid robot. In *Proc. of the RSS 2009 Workshop-Mobile Manipulation in Human Environments*.

Sinapov, J.; Schenck, C.; Staley, K.; Sukhoy, V.; and Stoytchev, A. 2014. Grounding semantic categories in behavioral interactions: Experiments with 100 objects. *Robotics and Autonomous Systems* 62(5):632–645.

Sinapov, J.; Khante, P.; Svetlik, M.; and Stone, P. 2016. Learning to order objects using haptic and proprioceptive exploratory behaviors. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*.

Sinapov, J.; Schenck, C.; and Stoytchev, A. 2014. Learning relational object categories using behavioral exploration and multimodal perception. In *IEEE International Conference on Robotics and Automation*, 5691–5698.

Sridharan, M.; Wyatt, J.; and Dearden, R. 2010. Planning to see: A hierarchical approach to planning visual actions on a robot using POMDPs. *Artificial Intelligence* 174(11):704–725.

Thomason, J.; Sinapov, J.; Svetlik, M.; Stone, P.; and Mooney, R. J. 2016. Learning multi-modal grounded linguistic semantics by playing I Spy. In *Proceedings of the Twenty-Fifth international joint conference on Artificial Intelligence*.

Whitney, D.; Eldon, M.; Oberlin, J.; and Tellex, S. 2016. Interpreting Multimodal Referring Expressions in Real Time. In *International Conference on Robotics and Automation*.

Zhang, S.; Khandelwal, P.; and Stone, P. 2017. Dynamically constructed (PO)MDPs for adaptive robot planning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 3855–3863.

Validation of Hierarchical Plans via Parsing of Attribute Grammars

Roman Barták, Adrien Maillard
Charles University
Faculty of Mathematics and Physics
Prague, Czech Republic

Rafael C. Cardoso
Pontifícia Universidade Católica do Rio Grande do Sul
Porto Alegre, Brazil

Abstract

An important problem of automated planning is validating if a plan complies with the planning domain model. Such validation is straightforward for classical sequential planning but until recently there was no such validation approach for Hierarchical Task Networks (HTN) planning. In this paper we propose a novel technique for validating HTN plans that is based on representing the HTN model as an attribute grammar and using a special parsing algorithm to verify if the plan can be generated by the grammar.

Introduction

Automated planning deals with the problem of finding a sequences of actions to reach a certain goal (Ghallab, Nau, and Traverso 2004). Actions are specified via preconditions and postconditions (also called effects) describing propositions that must be true in the state before action application (preconditions) and that will become true after action application (postconditions). Hence, action are a formal model of state transitions and a plan – a sequence of actions - describes a valid evolution of the world from a given initial state.

To increase efficiency of planning, Hierarchical Task Networks (HTN) were proposed to describe sets of actions as recipes for solving specific tasks (Erol, Hendler, and Nau 1996). HTN models are based on idea of decomposing compound tasks to subtasks until primitive tasks – actions – are obtained. The decomposition may include extra constraints describing precedence relations between sub-tasks and required properties of states (propositions that must hold before or between certain subtasks). The planning problem is specified as a goal task that needs to be decomposed to a sequence of actions applicable to an initial state, while satisfying all the task decomposition constraints and all the causal constraints between the actions. This sequence needs to be a valid plan in terms of causal constraints between the actions.

An important problem in automated planning is validating plans with respect to a given domain model. Such validation is easy for classical sequential planning, where it can be realised by simulating plan execution (Howey and Long 2003). However, until recently, there was no method to validate HTN plans, that is, to validate if a given plan can indeed be obtained from the goal task by some decomposition steps. There exists a recent validation method based on representing all possible decompositions as a SAT problem (Behnke, Höller, and Biundo 2017), but this method does not assume decomposition constraints (except decomposition preconditions that are compiled away to a dummy action). In this paper we suggest a more general approach that covers HTN models completely including all decomposition and causal constraints.

It has already been noted that derivation trees of Context-Free (CF) grammars resemble the structure of Hierarchical Task Networks (HTN). This has been used in (Erol, Hendler, and Nau 1996) to show the expressiveness of planning formalisms. Then, there have been some attempts to represent HTNs as CF grammars or equivalent formalisms (Nederhof, Shieber, and Satta 2003) but as demonstrated in (Höller et al. 2014), the languages defined by HTN planning problems (with partial-order, preconditions and effects) lie somewhere between CF and context-sensitive (CS) languages. In (Geib 2016), the author presents an approach with a similar intention with the help of *Combinatory Categorial Grammars* (CCGs), which are part of a category lying between CF and CS grammars, the *mildly context-sensitive grammars*. The author proposes a single model for both plan recognition and planning and he also proposes a planning algorithm based on CCGs. However, it appears that this modelling process is counter-intuitive as it requires a *lexicalization* (the hierarchical structure is contained in the terminal symbols) while the decomposition approach is more natural in planning. Also, it is not yet sure if this formalism and its planning technique can produce the full range of HTN plans. Recently, a model of HTNs based on attribute grammars has been proposed (Barták and Maillard 2017). The underlying grammar describes proper task decompositions, while a so called *timeline* constraint over the task attributes describes valid orders of actions based on causal relations. It is the only model that handles all HTN constraints including interleaving of actions. Though string shuffling used in plan recognition (Maraist 2017) allows some for of task interleaving, it is not clear how it maintains the causal constraints.

In this paper, we will use attribute grammars to validate HTN plans. We will describe how HTN domain model is represented as an attribute grammar, and for this grammar we will present a parsing technique that does plan validation. Note that due to interleaving of actions and presence of extra constraints, the parsing technique needs to be more general than classical parsing for CF grammars.

Background on Planning

In this paper we work with classical STRIPS planning that deals with sequences of actions transferring the world from a given initial state to a state satisfying certain goal condition. World states are modelled as sets of propositions that are true in those states and actions are changing validity of certain propositions.

Classical Planning

Formally, let P be a set of all propositions modelling properties of world states. Then a state $S \subseteq P$ is a set of propositions that are true in that state (every other proposition is false). Later, we will use the notation $S^+ = S$ to describe explicitly the valid propositions in the state S and $S^- = P \setminus S$ to describe explicitly the propositions that are not valid in the state S.

Each action a is described by four sets of propositions $(B_a^+, B_a^-, A_a^+, A_a^-)$, where $B_a^+, B_a^-, A_a^+, A_a^- \subseteq P, B_a^+ \cap B_a^- = \emptyset, A_a^+ \cap A_a^- = \emptyset$. Sets B_a^+ and B_a^- describe positive and negative preconditions of action a, that is, propositions that must be true and false right before the action a. Action a is applicable to state S iff $B_a^+ \subseteq S \wedge B_a^- \cap S = \emptyset$. Sets $A_a^+$and A_a^- describe positive and negative effects of action a, that is, propositions that will become true and false in the state right after executing the action a. If an action a is applicable to state S then the state right after the action a will be

$$\gamma(S, a) = (S \setminus A_a^-) \cup A_a^+. \tag{1}$$

If an action a is not applicable to state S then $\gamma(S, a)$ is undefined.

The classical planning problem, also called a STRIPS problem, consists of a set of actions A, a set of propositions S_0 called an initial state, and disjoint sets of goal propositions G^+ and G^- describing the propositions required to be true and false in the goal state. A solution to the planning problem is a sequence of actions $a_1, a_2, \ldots, a_n$ such that $S = \gamma(...\gamma(\gamma(S_0, a_1), a_2), ..., a_n)$ and $G^+ \subseteq S \wedge G^- \cap S = \emptyset$. This sequence of actions is called a *plan*.

Hierarchical Task Networks as Attribute Grammars

To simplify the planning process, several extensions of the basic STRIPS model were proposed to include some control knowledge. Hierarchical Task Networks (Erol, Hendler, and Nau 1996) were proposed as a planning domain modeling framework that includes control knowledge in the form of recipes how to solve specific tasks. The recipe is represented as a task network, which is a set of sub-tasks to solve a given task together with the set of constraints between the sub-tasks. The constraints can be of the following types:

- $t_1 \prec t_2$: a precedence constraint meaning that in every plan the last action obtained from task t_1 is before the first action obtained from task t_2,
- *before*(U, l): a precondition constraint meaning that in every plan the literal l holds in the state right before the first action obtained from tasks U,
- *after*(U, l): a postcondition constraint meaning that in every plan the literal l will hold in the state right after the last action obtained from tasks U,
- *between*(U, V, l): a prevailing condition meaning that in every plan the literal l holds in all the states between the last action obtained from tasks U and the first action obtained from tasks V.

In HTN, a compound task is solved by decomposing it to a task network - the connection between the task and the task network is called a (decomposition) *method*. The method can naturally be described as a rewriting rule of an attribute grammar. Attribute grammars (Knuth 1968) use the same type of rewriting rules as context-free grammars, but the grammar symbols may by annotated by attributes connected by constraints. This makes attribute grammars stronger than CF grammars in the sense of recognising a large class of languages.

Let $T(\overrightarrow{X})$ be a compound task with parameters $\overrightarrow{X}$ and $(\{T_1(\overrightarrow{X_1}), ..., T_k(\overrightarrow{X_k})\}, C)$ be a task network, where C are its constraints. We can encode the decomposition method as an attribute grammar rule:

$$T(\overrightarrow{X}) \rightarrow T_1(\overrightarrow{X_1}), ..., T_k(\overrightarrow{X_k}) \;\; [C] \tag{2}$$

The planning problem in HTN is specified by an initial state (the set of propositions that hold at the beginning) and by an initial task representing the goal. The compound tasks need to be decomposed via decomposition methods until a set of primitive tasks – actions – is obtained. Moreover, these actions need to be linearly ordered to satisfy all the constraints obtained during decompositions and the obtained plan – a linear sequence of actions – must be applicable to the initial state in the same sense as in classical planning.

If we do planning by application of grammar rewriting rules, we get a linear sequence of actions (a terminal word in terms of formal grammars), but this sequence does not necessarily form a valid plan as the actions from different tasks may interleave to satisfy the ordering and causal constraints (see Figure 1). So the actions obtained by applying the grammar rules need to be re-ordered to get a valid plan. The attribute grammars model the valid action orderings via a global timeline constraint (Barták and Maillard 2017).

To give a particular example of the decomposition rule, let us assume a task to transfer a container c from one location $l1$ to another location $l2$ by a robot r. To solve this task, we need to load the container first, then move it to its destination location, and unload it there. The following rule describes this decomposition method[1]:

$$\begin{aligned}\text{Transfer1}(c, l1, l2, r) \rightarrow\; & \text{Load-rob}(c, r, l1). \\ & \text{Move-rob}(r, l1, l2). \\ & \text{Unload-rob}(c, r, l2)[C]\end{aligned} \tag{3}$$

[1] There are several ways to model the task. For example, the *before* and *after* constraints can be omitted as they will be part of the primitive tasks.

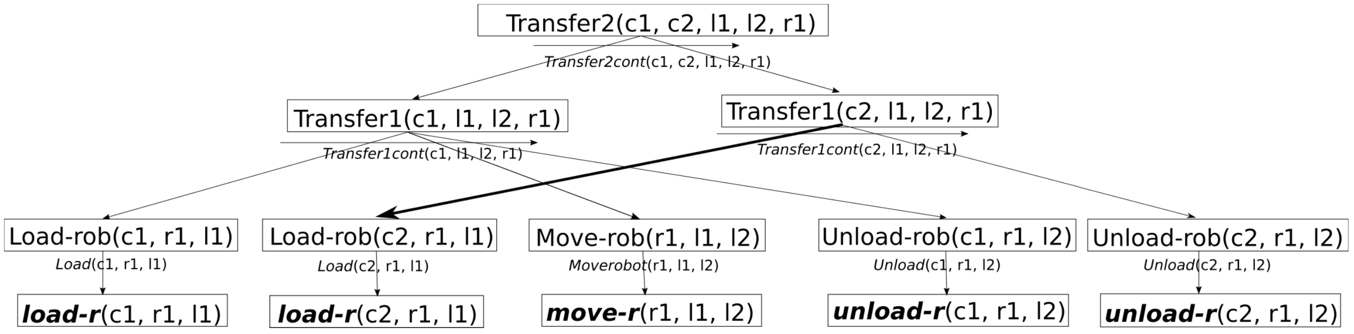


Figure 1: A task decomposition tree showing interleaving of actions obtained from decompositions of different tasks - denoted by the bold arc.

where

$$\begin{aligned} C = \{\, & \text{Load-rob} \prec \text{Move-rob},\ \text{Move-rob} \prec \text{Unload-rob}, \\ & \textit{before}(\{\text{Load-rob}\}, at(r, l1)), \\ & \textit{before}(\{\text{Load-rob}\}, at(c, l1)), \\ & \textit{between}(\{\text{Load-rob}\}, \{\text{Move-rob}\}, at(r, l1)), \\ & \textit{between}(\{\text{Move-rob}\}, \{\text{Unload-rob}\}, at(r, l2)), \\ & \textit{between}(\{\text{Load-rob}\}, \{\text{Unload-rob}\}, in(c, r)), \\ & \textit{after}(\{\text{Unload-rob}\}, at(c, l2)\} \end{aligned}$$

The decomposition constraints specify the following restrictions:

- the robot and the container must be at the same location $l1$ before loading,
- the robot does not change its location between loading and the start of moving,
- the container stays in the robot between loading and unloading,
- the robot stays at the destination location $l2$ between the end of moving and the start of unloading,
- the container will be at the destination location $l2$ after unloading.

An alternative decomposition method omits the Move-rob task as it assumes that this task is introduced by decomposition of another compound task. See the task for $c2$ in Figure 1. Still, we need to ensure that the robot is at the right location before unloading, which is done by the constraint *before*$(\{\text{Unload-rob}\}, at(r, l2))$. The alternative decomposition rule looks as follows:

$$\begin{aligned} \text{Transfer1}(c, l1, l2, r) \rightarrow & \text{Load-rob}(c, r, l1). \\ & \text{Unload-rob}(c, r, l2) \\ & [C] \end{aligned} \tag{4}$$

where

$$\begin{aligned} C = \{\, & \text{Load-rob} \prec \text{Unload-rob}, \\ & \textit{before}(\{\text{Load-rob}\}, at(r, l1)), \\ & \textit{before}(\{\text{Load-rob}\}, at(c, l1)), \\ & \textit{before}(\{\text{Unload-rob}\}, at(r, l2)), \\ & \textit{between}(\{\text{Load-rob}\}, \{\text{Unload-rob}\}, in(c, r)), \\ & \textit{after}(\{\text{Unload-rob}\}, at(c, l2)\} \end{aligned}$$

The top task for transferring two containers using the same robot and between the same locations can be described using the following decomposition method:

$$\begin{aligned} \text{Transfer2}(c1, c2, l1, l2, r) \rightarrow & \text{Transfer1}(c1, l1, l2, r). \\ & \text{Transfer1}(c2, l1, l2, r) \\ & [] \end{aligned} \tag{5}$$

Notice that having the $before$ and $after$ constraints allows us to describe action preconditions and postconditions as decomposition constraints rather than having them specified separately. This is done by having a compound task for each action, for example Load-rob corresponds to the primitive action load-r. This is the corresponding decomposition method:

$$\text{Load-rob}(c, r, l) \rightarrow \text{load-r}(c, r, l). \qquad [C] \tag{6}$$

where

$$\begin{aligned} C = \{ & \textit{before}(\{\text{load-r}\}, at(r, l)), \\ & \textit{before}(\{\text{load-r}\}, at(c, l)), \\ & \textit{after}(\{\text{load-r}\}, in(c, r)\} \\ & \textit{after}(\{\text{load-r}\}, \neg at(c, l)\} \end{aligned}$$

HTN Validation Algorithm

The plan validation problem is a problem reverse to the planning problem. We have a plan as the input and the problem is to validate if that plan can be obtained by decomposition from the goal task. In terms of grammars, it means using the grammar rules in an analytical way to do parsing.

Recall that the order of actions in the plan does not necessarily correspond to the order of actions obtained by application of grammar rules. Hence, during parsing, we ignore the order of tasks on the right side of grammar rules and we model the action (task) order explicitly by using indexes assigned to tasks. Each task will be annotated by two indexes describing the order numbers of the first and the last actions obtained from task decomposition. For example, the task $\text{Load-rob}_{1,1}(c1, r1, l1)$ from Figure 1, that gives the action $\text{load-r}(c1, r1, l1)$, is annotated by indexes 1,1.

Let us now demonstrate a single parsing step. Assume that we already parsed the tasks $\text{Load-rob}_{1,1}(c1, r1, l1)$,

$\text{Move-rob}_{3,3}(r1, l1, l2)$, and $\text{Unload-rob}_{4,4}(c1, r1, l2)$ and we continue in parsing using the grammar rule (3). The tasks on the right side of the rule already exist and we can verify the ordering constraints $1 \prec 3$ and $3 \prec 4$ by comparing the respective indexes. The result of the parsing step will be a new parsed task $\text{Transfer1}_{1,4}(c1, l1, l2, r1)$, where the indexes are taken as minimal and maximal indexes of its subtasks.

We still need to verify the other constraints in the rule. This will be done by maintaining a timeline for each task. The *timeline* is a sequence of slots describing validity of literals in time steps corresponding to the task. For every time step, the slot will describe the literals that hold in the state before the action at that time (a Pre part) and literals that must hold in the state right after the action (a Post part). For example, the task $\text{Load-rob}_{1,1}(c1, r1, l1)$ will use a single slot $(\{at(r1, l1), at(c1, l1)\}, \{in(c1, r1), \neg at(c1, l1)\})_1$, where the index represents time and the literals are basically preconditions and postconditions of action $\text{load-r}(c1, r1, l1)$ that were encoded as *before* and *after* constraints (see the rule (6)).

During the parsing step, we first *merge* the timelines for the subtasks with possible insertion of empty slots for times not covered by the sub-tasks (slot 2 in our example). Empty slot does not contain any action, but its Pre part may contain literals obtained by propagation (see below). Two slots with the same index can only be merged if (at least) one of them is empty. This way we ensure that each action is generated exactly once. For example, when merging timelines for tasks $\text{Transfer1}_{1,4}(c1, l1, l2, r1)$ and $\text{Transfer1}_{2,5}(c2, l1, l2, r1)$ we are merging non-empty slots 1,3,4 for the first task with non-empty slots 2, 5 of the second task. If the slots cannot be merged as they both already contain an action, then processing of the derivation rule is stopped and the algorithm continues with the next rule.

After merging the timelines for subtasks we add literals based on the rule *constraints* - for *before* and *between* constraints, the literals are added to the Pre parts of respective slots; for the *after* constraints, the literals are added to the Post parts.

After that, we *propagate* the literals between the slots. This propagation goes from left to right, where the literals from the postcondition part are added to the precondition part of the next slot and, if the slot is not empty (contains some action), the literals in preconditions, that are not deleted by the action, are added to the precondition part of the next slot. This basically follows the state transition formula as specified in (1). The right-to-left propagation adds literals in preconditions to preconditions of the previous slot provided that the slot is not empty and the literal is not added by the action in it. The goal of propagation is to keep information about states up-to-date (notice that propagation changes only the Pre parts of the slots that describe the states).

Finally, we verify that the slots are consistent, which consists of checking that no slot contains a literal and its negation in any of its parts. Table 1 demonstrates this process – it shows how literals are added to the slots in each step (slot merging, constraint addition, propagation).

The validation algorithm first transfers each action to a primitive task with the index corresponding to the order of the action and with the timeline containing a single slot with that action and empty Pre and Post parts. Recall, that preconditions and postconditions of actions will be added later during parsing using the rules of type (6). The literals of the initial state are added to the Pre part of the first slot (for simplicity, we ignored them in the previous example of a parsing step). Then the algorithm takes any grammar rule such that the tasks from its right side are already known and it does the above described parsing step. This may introduce a new parsed task. This process is repeated while some new task is introduced or until a goal task is introduced whose indexes span the whole plan. If the goal task is found then the plan is sound, otherwise, the plan is not sound. Note that the algorithm always finishes as there is only a finite number of compound tasks that can introduced during parsing. We will now describe the validation algorithm formally.

Data structures

First we will describe the data structures that are used later in the algorithm. Basically, we will introduce slots, timelines, and the parsed tasks :

We define the type `slot` as a tuple $(\text{Pre}^+, \text{Pre}^-, a, \text{Post}^+, \text{Post}^-)$ where

- Pre^+ is a set of atoms (positive propositions in the state)
- Pre^- is a set of atoms (negative propositions in the state)
- $a \in A \cup \{\textit{empty}\}$ is an action name (or an empty slot)
- Post^+ is a set of atoms (positive postconditions of a)
- Post^- is a set of atoms (negative postconditions of a)

To simplify verification of slot/timeline soundness we use separate sets for positive and negative propositions. Note also that the sets $\text{Pre}^+, \text{Pre}^-$ are not only related to action a but they will describe the state right before the action. More precisely, these sets describe the propostions that must hold in the state, but until all slots are non-empty, the state may be described only partially (see Table 1).

Then, we define the type `subplan` that represents a parsed task T as a tuple $(T, b, e, \textit{timeline})$ with

- T being a task name,
- b and e ($b \leq e$) being two integers equal to the indexes in the original plan of the first and last actions in the subplan generated from T; this pair shows how much the subplan generated from T *spans over* the verified plan,
- $\textit{timeline}$ being an ordered sequence of $(e - b + 1)$ elements of the `slot` type; we have $\textit{timeline} = \{s_b, ..., s_e\} \subseteq \textbf{slots}$.

The algorithm formally

The validation algorithm is shown in Algorithm 1. At the beginning, actions in the plan are put individually in the set $\textbf{subplans}$ (line 2). They are all subplans of size 1. The initial state is added to the Pre parts of the slot of the first action. Then, at each iteration the algorithm fires rules in the grammar where all subtasks are elements of $\textbf{subplans}$. When

Table 1: The process of building a timeline during parsing the compound task $\mathrm{Transfer1}_{1,4}(c1,l1,l2,r1)$.

	1: load-r$(c1,r1,l1)$		2: *empty*		3: move-r$(r1,l1,l2)$		4: unload-r$(c1,r1,l2)$	
	Pre_1	$Post_1$	Pre_2	$Post_2$	Pre_3	$Post_3$	Pre_4	$Post_4$
merge	$at(r1,l1)$ $at(c1,l1)$	$in(c1,r1)$ $\neg at(c1,l1)$			$at(r1,l1)$	$\neg at(r1,l1)$ $at(r1,l2)$	$in(c1,r1)$ $at(r1,l2)$	$\neg in(c1,r1)$ $at(c1,l2)$
constrain			$at(r1,l1)$ $in(c1,r1)$		$in(c1,r1)$			
propagate			$\neg at(c1,l1)$				$\neg at(r1,l1)$	

such a rule is found, the precedence constraints are checked (line 7). Then the timelines of subtasks are merged (line 8) and before, after, and between constraints from the grammar rule are applied to this merged timeline (lines 9, 10, and 11). Preconditions and postconditions are then propagated from left to right and from right to left (line 12). Finally, the resulting timeline is verified (13). If no inconsistency is detected, then the new parsed task is added to the set $\mathbf{subplans}$ so it can be further used for building a higher-level task. Inconsistency means that some atom is both in the positive and in the negative parts of the state.

The positive exit condition (cf. Algorithm 2) is met when there is a *Goal* task in $\mathbf{subplans}$ that contains all the elements of the verified plan $\mathbf{P}$.

If, it is not possible to find a rule that applies to the current elements of $\mathbf{subplans}$ and produces a *new* subplan, then it means that the plan $\mathbf{P}$ is not valid with regard to the grammar. In other words, the set $\mathbf{subplans}$ has not grown during the execution of the for-loop (lines 6 to 18). At this point, the algorithm returns false (line 20).

We also include all the sub-procedures for merging the timelines and for applying the constraints. To simplify notations in the procedures for constraint application (Algorithms 5-7), we use the following notation – if l is a positive literal p then $l^+ = \{p\}$ and $l^- = \{\}$; if l is a negative literal $\neg p$ then $l^+ = \{\}$ and $l^- = \{p\}$.

Soundness

We shall now show that the algorithm correctly recognises plans that can be derived from a given *Goal* task and an initial state.

First, one should realise that the algorithm always finishes. All sub-procedures clearly finish as they consist of *for* loops and *if-then-else* conditions only. During each iteration of the main *while* loop, some new task may be added to the set of $\mathbf{subplans}$. The input plan is finite and we have only a finite number of constants so the number of tasks that can be derived is obviously finite. Hence the *while* loop must finish sometime, either when no new task is added (line 20) or when the *Goal* task is derived (line 5).

Assume that the algorithm finished successfully (with the answer **true**). It means that it found the *Goal* task that spans over the full plan (test in Algorithm 2). By reconstructing how this task was added to the set $\mathbf{subplans}$, we get the derivation tree (such as the one in Figure 1). We indeed get a tree as during merging of timelines, two slots can only be merged if at least one of them is empty. Hence each task in the tree has exactly one parent. If the same task appears two (or more) times in the tree then its slots would eventually merge with themselves, which is not possible (see Algorithm 4). All the constraints used in this derivation (decomposition) are satisfied as the algorithm verified the precedence constraints and added the literals from the before, after, and between constraints to the timeline, which is consistent.

Notice that the $Post$ parts of the slots in the timeline contain only the propositions from the after constraints so they model the effects of actions. The Pre parts (in particular the Pre^+ sets) model the states between the actions and we shall show that the sequence of states is correct with respect to the plan. First, each state is sound as it does not contain an atom and its negation ($\mathrm{Pre}^+ \cap \mathrm{Pre}^- = \emptyset$). Next, two subsequent states Pre_i^+ and Pre_{i+1}^+ model a correct state transition thanks to the propagation:

$$\mathrm{Pre}_{i+1}^+ = (\mathrm{Pre}_i^+ \setminus \mathrm{Post}_i^-) \cup \mathrm{Post}_i^+$$

$$\mathrm{Pre}_{i+1}^- = (\mathrm{Pre}_i^- \setminus \mathrm{Post}_i^+) \cup \mathrm{Post}_i^-$$

This realises the state transition formula (1). We will show it for the positive part of the state (the proof is identical for the negative part). Assume slots i and $i+1$ with some action filled in the slot i (the action must appear there eventually as the final timeline has all slots non-empty). Thanks to left-to-right propagation, it must hold $\mathrm{Post}_i^+ \subseteq \mathrm{Pre}_{i+1}^+$ (line 5 of Algorithm 8) and $\mathrm{Pre}_i^+ \setminus \mathrm{Post}_i^- \subseteq \mathrm{Pre}_{i+1}^+$ (line 8 of Algorithm 8). Thanks to right-to-left propagation, it must hold $\mathrm{Pre}_{i+1}^+ \setminus \mathrm{Post}_i^+ \subseteq \mathrm{Pre}_i^+$ (line 14 of Algorithm 8). It means that if a proposition $p \in \mathrm{Pre}_{i+1}^+$ is not added by the action ($p \notin \mathrm{Post}_i^+$) then p must already be part of the previous state ($p \in \mathrm{Pre}_i^+$). Together, we get:

$$\mathrm{Pre}_{i+1}^+ = (\mathrm{Pre}_i^+ \setminus \mathrm{Post}_i^-) \cup \mathrm{Post}_i^+ \qquad (7)$$

Notice that the algorithm works even when no initial state is provided. Then the final sets Pre_1^+ and Pre_1^- specify the propositions that must and must not be valid at the beginning to have a valid plan. If the initial state is provided then it is propagated through the slots.

In summary, the set of actions in the plan is generated by the grammar and forms a valid plan.

If the algorithm finishes with the answer **false** then no derivation exists as no other task can be parsed. Being the plan correct, the derivation tree would be reconstructed by the algorithm as the algorithm finds all the tasks that decompose to any subset of the plan.

```
Data: a plan $\mathbf{P} = (a_1, ..., a_n)$, initial state InitState, a goal
      task Goal, an attribute grammar
      $G = (\Sigma, N, \mathcal{P}, S, A, C)$
Result: a boolean equal to true if the plan can be derived
        from the hierarchical structure, false otherwise
Function VERIFYPLAN
    /* Initialization of the set of
       subplans                                  */
    $\mathbf{subplans} \leftarrow \{(a_i, i, i, \{(\emptyset, \emptyset, a_i, \emptyset, \emptyset)_i\}) | a_i \in \mathbf{P}\}$ ;
    $\mathrm{Pre}_1^+ \leftarrow InitState^+$;
    $\mathrm{Pre}_1^- \leftarrow InitState^-$;
    while $\neg$PLANISVALID$(\mathbf{subplans}, \mathbf{P}, Goal)$ do
        for each rule $R$ in $\mathcal{P}$ of the form
          $T_0 \rightarrow T_1, ..., T_k\ [\prec, pre, post, btw]$ such that
          $subtasks = \{(T_i, b_i, e_i, tl_i) | i \in 1..k\} \subseteq$
          $\mathbf{subplans}$ do
            verify $\prec$ from rule $R$ else break;
            $timeline \leftarrow$ MERGEPLANS$(subtasks)$;
            APPLYPRE$(timeline, pre)$;
            APPLYPOST$(timeline, post)$;
            APPLYBETWEEN$(timeline, btw)$;
            PROPAGATE$(timeline)$;
            if $\exists(\mathrm{Pre}^+, \mathrm{Pre}^-, a, \mathrm{Post}^+, \mathrm{Post}^-) \in$
              $timeline, \mathrm{Pre}^+ \cap \mathrm{Pre}^- \neq$
              $\emptyset \vee \mathrm{Post}^+ \cap \mathrm{Post}^- \neq \emptyset$ then
                break
            end
            $b = \min_{(T_i,b_i,e_i,tl_i)\in subtasks} b_i$,;
            $e = \max_{(T_i,b_i,e_i,tl_i)\in subtasks} e_i$;
            $\mathbf{subplans} \leftarrow$
              $\mathbf{subplans} \cup \{(T_0, b, e, timeline)\}$;
        end
        if size of subplans has not increased since the
          last iteration then
            return false
        end
    end
    return true
end
```

Algorithm 1: Verification procedure

```
Data: the set of subplans: subplans, the plan to be
      validated $\mathbf{P}$, the goal task $Goal$
Result: true or false
Function PLANISVALID
    return $(\exists(Goal, 1, |\mathbf{P}|, timeline) \in$
      $\mathbf{subplans}, s.t. \bigcup_{(\_,\_,a_i,\_,\_)\in timeline}\{a_i\} = \mathbf{P})$
end
```

Algorithm 2: The end condition of the valid plan

```
Data: a set of subplans : $subplans$
Result: a set of slots $newtimeline$, the aggregation of the
        slots of every subplan
Function MERGEPLANS$(subplans)$
    $lb = \min_{(T_i,b_i,e_i,timeline_i)\in subplans} b_i$;
    $ub = \max_{(T_i,b_i,e_i,timeline_i)\in subplans} e_i$;
    $newtimeline \leftarrow \{(\emptyset, \emptyset, empty, \emptyset, \emptyset)_i | i \in lb..ub\}$;
    for $(T, b, e, timeline) \in subplans$ do
        for $s_k \in timeline, s_k^{'} \in newtimeline$ do
            $s_k^{'} \leftarrow$ MERGESLOTS$(s_k, s_k^{'})$
        end
    end
    return $newtimeline$
end
```

Algorithm 3: Merge timelines

We showed that the algorithm always finishes. If it returns **true** then the plan can be derived from the *Goal* task. If it returns **false** then the plan cannot be derived from the *Goal* task. Hence the algorithm validates the plans with respect to the domain model.

Initial Experiments

In this section we report some initial experiments comparing the performance of the implementation of our algorithm against the PANDA verifier, described in (Behnke, Höller, and Biundo 2017). The PANDA verifier validates a plan by translating it into a SAT formula. This translation requires a bound, the maximum height of the decomposition that any candidate for a solution plan can have.

In these experiments we use the Transport domain, initially introduced in the International Planning Competition (IPC) of 2008, but without action costs. In this domain, each vehicle can transport packages between different locations based on road connections. Our implementation is able to parse directly from SHOP2 planner's (Nau et al. 2003) input files, arguably one of the most used HTN planner. At the moment, we only support basic HTN syntax from SHOP2, but we are gradually adding support for many SHOP2 commands and tags. PANDA verifier uses its own input, which is a PDDL-like representation of HTN.

Our Transport domain description in SHOP2 syntax contains three primitive tasks and three non-primitive tasks. The description used in PANDA verifier has four primitive tasks and six non-primitive tasks. The extra primitive task is a *noop* action, which in our description is encoded directly as a non-primitive task. The extra non-primitive tasks from PANDA's description are dummy methods that represent primitive tasks.

We ran 5 different problem instances and collected the total CPU times. These times include any parsing done by both approaches, and was calculated from the start to the end of each validation. To run these experiments we used a virtual machine (Oracle VM VirtualBox Version 5.1.22) running an Ubuntu 16.04 LTS, with 4 GB of memory and an Intel Core i7-4700MQ processor with 4 cores and 8 threads. Our implementation requires Ruby (we used version 2.3.1), while the PANDA verifier requires Java (we used OpenJDK 1.8) and the MiniSat solver (we used version 2.2.1).

Table 2 shows the initial results comparing our attribute grammar approach with PANDA verifier using the transport domain (with no action cost). The first problem instance ($p1$) has a solution plan with 8 actions, and an initial state with 15 ground atoms. Each subsequent problem instance has the following number of actions and number of ground atoms: 12 and 29; 16 and 45; 19 and 60; 22 and 80. Odd problems ($p1$, $p3$, and $p5$) had valid solutions, while even problems ($p2$, and $p4$) had not.

```
Data: two slots
      $s_1 = (\text{Pre}_1^+, \text{Pre}_1^-, a_1, \text{Post}_1^+, \text{Post}_1^-)$, $s_2 =$
      $(\text{Pre}_2^+, \text{Pre}_2^-, a_2, \text{Post}_2^+, \text{Post}_2^-)$
Result: merged slots
Function MERGESLOTS($s_1, s_2$)
    if $a_1 = empty$ or $a_2 = empty$ then
        $\text{Pre}^+ = \text{Pre}_1^+ \cup \text{Pre}_2^+$;
        $\text{Pre}^- = \text{Pre}_1^- \cup \text{Pre}_2^-$;
        $\text{Post}^+ = \text{Post}_1^+ \cup \text{Post}_2^-$;
        $\text{Post}^- = \text{Post}_1^- \cup \text{Post}_2^-$;
        $a = a_1(if\ a_2 = empty)\ or\ a_2(if\ a_1 = empty)$;
        return $(\text{Pre}^+, \text{Pre}^-, a, \text{Post}^+, \text{Post}^-)$
    end
    break
end
```

Algorithm 4: Merge slots

```
Data: a set of slot : $slots$, a set of $before$ constraints
Result: an updated set of slots
Function APPLYPRE($slots, pre$)
    for $before(U, l) \in pre$ do
        $id = \min\{b_i | T_i \in U\}$;
        $\text{Pre}_{id}^+ \leftarrow \text{Pre}_{id}^+ \cup l^+$;
        $\text{Pre}_{id}^- \leftarrow \text{Pre}_{id}^- \cup l^-$
    end
end
```

Algorithm 5: Apply before constraints

For these initial experiments, our approach appear to scale linearly when the solution is valid, but takes a bit more time if it is not valid, as shown in Figure 2. PANDA verifier had an exception on $p2$, because it does not seem to allow invalid transitions, but instead of ignoring that decomposition path, it crashes with an exception. And in $p5$, Panda returned that the plan was not valid, which was incorrect.

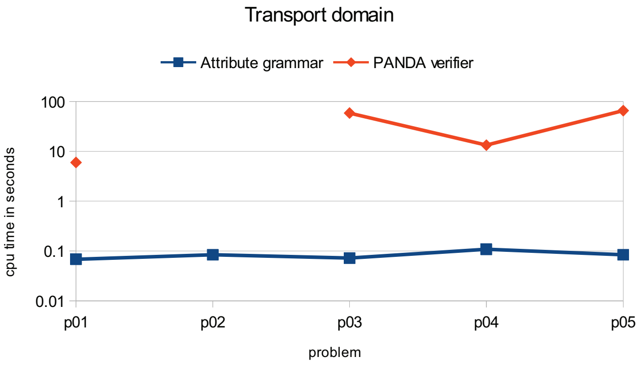


Figure 2: Transport domain results.

Conclusions

In this paper we proposed an algorithm for validating HTN plans by using parsing of an attribute grammar describing the HTN domain model. The algorithm mimics classical parsing of context-free grammars customised to attribute

```
Data: a set of slot : $slots$, a set of $after$ constraints
Result: an updated set of slots
Function APPLYPOST($slots, post$)
    for $after(U, l) \in post$ do
        $id = \max\{e_i | T_i \in U\}$;
        $\text{Post}_{id}^+ \leftarrow \text{Post}_{id}^+ \cup l^+$;
        $\text{Post}_{id}^- \leftarrow \text{Post}_{id}^- \cup l^-$
    end
end
```

Algorithm 6: Apply after constraints

```
Data: a set of slot : $slots$, a set of $between$ constraints
Result: an updated set of slots
Function APPLYBETWEEN($slots, between$)
    for $between(U, V, l) \in between$ do
        $s = \max\{e_i | T_i \in U\} + 1$;
        $e = \min\{b_i | T_i \in V\}$;
        for $id = s$ to $e$ do
            $\text{Pre}_{id}^+ \leftarrow \text{Pre}_{id}^+ \cup l^+$;
            $\text{Pre}_{id}^- \leftarrow \text{Pre}_{id}^- \cup l^-$
        end
    end
end
```

Algorithm 7: Apply between constraints

grammars with the timeline constraint.

The algorithm starts with the plan and applies the decomposition rules in a reverse order to group actions into tasks. The decomposition constraints are verified by keeping information about propositions that must be true at states before and after actions. The algorithm stops when it finds a task that covers the complete plan. Then the plan is valid. The other way of stopping the algorithm is when no other compound task can be constructed. In such a case the plan does not correspond to any task. Note, that the plan might still be a correct sequence of actions but it cannot be obtained by decomposition of any task.

The major innovation of the proposed technique is that it is the first approach that covers HTN models fully including interleaving of actions and various decomposition constraints. In particular, the proposed algorithm is more general than an existing SAT-based approach (Behnke, Höller, and Biundo 2017) in covering precedence, before, between, and after constraints. The SAT-based approach only covers specific before constraints (the constraint is applied to the set of all tasks on the right side of the rule) that must be encoded as dummy actions. These dummy actions must be part of the plan to be validated so for the original plan to be validated one must find proper places, where to insert these dummy actions, which is not discussed in (Behnke, Höller, and Biundo 2017).

Furthermore, our initial experiments indicate that converting HTN models to attribute grammars may provide better performance results for validating plans, rather than converting to SAT. More experiments with other domains are needed to ascertain in which types of domain each approach performs better.

As other planning models such as procedural domain con-

```
Data: a set of slots $slots$
Result: an updated set of slots
Function PROPAGATE($slots$)
    $lb = \min_{(\mathrm{Pre}_j^+,\mathrm{Pre}_j^-,a_j,\mathrm{Post}_j^+,\mathrm{Post}_j^-)\in slots} j$;
    $ub = \max_{(\mathrm{Pre}_j^+,\mathrm{Pre}_j^-,a_j,\mathrm{Post}_j^+,\mathrm{Post}_j^-)\in slots} j - 1$;
    /* Propagation to the right */
    for $i = lb$ to $ub$ do
        $\mathrm{Pre}_{i+1}^+ \leftarrow \mathrm{Pre}_{i+1}^+ \cup \mathrm{Post}_i^+$;
        $\mathrm{Pre}_{i+1}^- \leftarrow \mathrm{Pre}_{i+1}^- \cup \mathrm{Post}_i^-$;
        if $a_i \neq$ empty then
            $\mathrm{Pre}_{i+1}^+ \leftarrow \mathrm{Pre}_{i+1}^+ \cup (\mathrm{Pre}_i^+ \setminus \mathrm{Post}_i^-)$;
            $\mathrm{Pre}_{i+1}^- \leftarrow \mathrm{Pre}_{i+1}^- \cup (\mathrm{Pre}_i^- \setminus \mathrm{Post}_i^+)$
        end
    end
    /* Propagation to the left */
    for $i = ub$ downto $lb$ do
        if $a_i \neq$ empty then
            $\mathrm{Pre}_i^+ \leftarrow \mathrm{Pre}_i^+ \cup (\mathrm{Pre}_{i+1}^+ \setminus \mathrm{Post}_i^+)$;
            $\mathrm{Pre}_i^- \leftarrow \mathrm{Pre}_i^- \cup (\mathrm{Pre}_{i+1}^- \setminus \mathrm{Post}_i^-)$
        end
    end
end
```

Algorithm 8: Propagate

Table 2: Initial results of experiments comparing CPU run time, in seconds.

transport domain	**p01**	**p02**	**p03**	**p04**	**p05**
	CPU time	CPU time	CPU time	CPU time	CPU time
Attribute grammar	0.068	0.084	0.072	0.108	0.084
PANDA verifier	5.968	-	58.52	13.32	65.56 wrong

trol knowledge (Baier, Fritz, and McIlraith 2007) can be translated to attribute grammars (Barták and Maillard 2017) the proposed algorithm can verify plans with respect to these models too.

Our current implementation of the algorithm uses a straightforward approach to find rules used for parsing. The more efficient implementation of the algorithm may exploit principles of the Rete algorithm (Forgy 1982) used for production rule systems.

Acknowledgments

Research is supported by the Czech Science Foundation under the project P103-15-19877S.

References

Baier, J. A.; Fritz, C.; and McIlraith, S. A. 2007. Exploiting Procedural Domain Control Knowledge in State-of-the-Art Planners. In Boddy, M. S.; Fox, M.; and Thiébaux, S., eds., *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling, ICAPS 2007, Providence, Rhode Island, USA, September 22-26, 2007*, 26–33. AAAI.

Barták, R., and Maillard, A. 2017. Attribute grammars with set attributes and global constraints as a unifying framework for planning domain models. In *Proc. of the ICAPS Workshop on Knowledge Engineering for Planning and Scheduling, KEPS 1017*, 45–53.

Behnke, G.; Höller, D.; and Biundo, S. 2017. This is a solution! (... but is it though?) verifying solutions of hierarchical planning problems. In *Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling, ICAPS 2017)*, 20–28.

Erol, K.; Hendler, J. A.; and Nau, D. S. 1996. Complexity Results for HTN Planning. *Ann. Math. Artif. Intell.* 18(1):69–93.

Forgy, C. L. 1982. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence* 19(1):17 – 37.

Geib, C. 2016. Lexicalized reasoning about actions. *Advances in Cognitive Systems* 4:187–206.

Ghallab, M.; Nau, D. S.; and Traverso, P. 2004. *Automated planning - theory and practice*. Elsevier.

Howey, R., and Long, D. 2003. VAL's Progress: The Automatic Validation Tool for PDDL2.1 used in the International Planning Competition. In *Proceedings of ICAPS'03 Workshop on the Competition: Impact, Organization, Evaluation, Benchmarks*.

Höller, D.; Behnke, G.; Bercher, P.; and Biundo, S. 2014. Language Classification of Hierarchical Planning Problems. In Schaub, T.; Friedrich, G.; and O'Sullivan, B., eds., *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, 447–452. IOS Press.

Knuth, D. E. 1968. Semantics of Context-Free Languages. *Mathematical Systems Theory* 2(2):127–145.

Maraist, J. 2017. String shuffling over a gap between parsing and plan recognition. In *The AAAI-17 Workshop on Plan, Activity, and Intent Recognition WS-17-13*, 835–842.

Nau, D.; Ilghami, O.; Kuter, U.; Murdock, J. W.; Wu, D.; and Yaman, F. 2003. Shop2: An htn planning system. *Journal of Artificial Intelligence Research* 20:379–404.

Nederhof, M.-J.; Shieber, S.; and Satta, G. 2003. Partially ordered multiset context-free grammars and ID/LP parsing. *Proceedings of the Eighth International Workshop on Parsing Technologies* 171–182.

Situated Planning for Execution Under Temporal Constraints

Michael Cashmore, Andrew Coles
King's College London

Bence Cserna
University of New Hampshire

Erez Karpas
Technion — Israel Institute of Technology

Daniele Magazzeni
King's College London

Wheeler Ruml
University of New Hampshire

Abstract

One of the original motivations for domain-independent planning was to generate plans that would then be executed in the environment. However, most existing planners ignore the passage of time during planning. While this can work well when absolute time does not play a role, this approach can lead to plans failing when there are external timing constraints, such as deadlines. In this paper, we describe a new approach for time-sensitive temporal planning. Our planner is aware of the fact that plan execution will start only once planning finishes, and incorporates this information into its decision making, in order to focus the search on branches that are more likely to lead to plans that will be feasible when the planner finishes.

Introduction

One of the original motivations for domain-independent planning was for controlling robots performing complex tasks (Fikes and Nilsson 1971). The typical approach to controlling robots using a planner is to call the planner to generate a plan which solves the problem, and then execute that plan in the environment. This approach works well if the plan remains applicable regardless of when it is executed. However, if there are external timing constraints, such as deadlines which must be met, things become more complex. This is because we must take into account the *planning time*.

For example, in the Robocup Logistics League (RCLL) challenge (Niemueller, Lakemeyer, and Ferrein 2015), a team of robots must move workpieces between different machines that perform some operations on them, and fulfill some order with a deadline. This calls for using temporal planning, because we would like all robots to work in parallel, and actions have different durations. The typical approach would have the planner come up with a plan which would work had it been executed at time 0, and then execute this plan when the planner completes. Obviously, this might lead to missing the deadline, and thus, plan failure.

One simple approach to handling this problem is to use some estimate on how long planning will take, and adapt all the deadlines assuming plan execution would start when the planner finishes. While using an upper bound on planning time will eliminate the problem of plans failing, it might lead to the planner not finding a feasible plan to begin with. On the other hand, using too low an estimate could still lead to plans failing, as discussed above.

In this paper, we describe a new approach for situated temporal planning. Our planner is aware of the fact that plan execution will start once planning finishes, and incorporates this information into the internal data structure for temporal reasoning used by the planner, together with estimates of remaining planning time. This helps our planner prune partial plans which are likely to lead to the planner finishing planning too late for the plans to be of use, and focus on more promising branches of the search.

Our empirical evaluation demonstrates that this planner can handle temporal planning problems with absolute deadlines much better than a naive baseline approach, in realistic settings where planning time counts, and the plan can only start executing once it is completed. To the best of our knowledge, this is the first temporal planner to explicitly consider planning time, within the context of planning and execution. Thus, our planner is especially applicable to online planning for robotics, where a robot must find a plan to execute, but the world does not stop while the robot is planning.

Preliminaries

We consider propositional temporal planning problems with Timed Initial Literals (TIL) (Cresswell and Coddington 2003; Edelkamp and Hoffmann 2004). Such a planning problem Π is specified by a tuple $\Pi = \langle F, A, I, T, G \rangle$, where:

- F is a set of Boolean propositions, which describe the state of the world.
- A is a set of durative actions. Each action $a \in A$ is described by:
  - Minimum duration $dur_{\min}(a)$ and maximum duration $dur_{\max}(a)$, both in $\mathbb{R}^{0+}$ with $dur_{\min}(a) \leq dur_{\max}(a)$,
  - Start condition $cond_{\vdash}(a)$, invariant condition $cond_{\leftrightarrow}(a)$, and end condition $cond_{\dashv}(a)$, all of which are subsets of F, and
  - Start effect $\mathit{eff}_{\vdash}(a)$ and end effect $\mathit{eff}_{\dashv}(a)$, both of which specify which propositions in F become true (add effects), and which become false (delete effects).
- $I \subseteq F$ is the initial state, specifying exactly which propositions are true at time 0.

- T is a set of timed initial literals (TIL). Each TIL $l \in T$ consists of a time $time(l)$ and a literal $lit(l)$, which specifies which proposition in F becomes true (or false) at time $time(l)$.
- $G \subseteq F$ specifies the goal, that is, which propositions we want to be true at the end of plan execution.

A solution to a temporal planning problem is a schedule σ, which is a sequence of triples $\langle a, t, d \rangle$, where $a \in A$ is an action, $t \in \mathbb{R}^{0+}$ is the time when action a is started, and $d \in [dur_{\min}(a), dur_{\max}(a)]$ is the duration chosen for a. A schedule can be seen as a set of instantaneous *happenings* (Fox and Long 2003), which occur when an action starts, when an action ends, and when a timed initial literal is triggered. Specifically, for each triple $\langle a, t, d \rangle$ in the schedule, we have action a starting at time t (requiring $cond_{\vdash}(a)$ to hold a small amount of time ϵ before time t, and applying the effects $\mathit{eff}_{\vdash}(a)$ right at t), and ending at time $t + d$ (requiring $cond_{\dashv}(a)$ to hold ϵ before $t + d$, and applying the effects $\mathit{eff}_{\dashv}(a)$ at time $t + d$). For a TIL l we have the effect specified by $lit(l)$ triggered at time $time(l)$. Finally, in order for a schedule to be valid, we also require the invariant condition $cond_{\leftrightarrow}(a)$ to hold over the open interval between t and $t+d$, and that the goal G holds at the state which holds after all happenings have occurred.

Related Work

Temporal planners have of course been used in on-line applications before. For example, researchers at PARC built a special-purpose temporal planner for on-line manufacturing (Ruml et al. 2011). As in many temporal planners, each search node contains a Simple Temporal Network (STN) (Dechter, Meiri, and Pearl 1991) to represent the time points of events in the plan and constraints on when they can occur. To reflect the fact that actions cannot occur until planning has completed, the PARC planner includes a hard-coded estimate of the required planning time, and every time point in the STN is constrained to occur at least that far after the time that planning started (Ruml et al. 2011, Figure 11). While this is a reasonable solution in a domain where the expected planning problems are all of similar difficulty, this approach can perform poorly in domains that include a wide variety of problems, as we will see below.

There has also been work on time-aware planning in the search community. Dionne, Thayer and Ruml (2011) present a so-called 'contract algorithm' called Deadline-Aware Search (DAS) that, given a deadline, attempts to return the cheapest complete plan that it can find within that deadline. The main part of the algorithm works by estimating the time that will be required to find a solution beneath each node in the open list, and pruning those for which this estimate exceeds the remaining search time. The estimate is the product of three quantities that are determined on-line: the time required to expand a node, expressed in seconds, an estimate on the number of search nodes remaining on the path to a goal beneath the given node, notated $d(n)$, and the average number of expansions required before a generated node is selected for expansion, called the *expansion delay*. Although DAS was shown to surpass anytime algorithms on combinatorial benchmarks, its ideas have never been implemented in a domain-independent planner.

Bugsy (Burns, Ruml, and Do 2013) is a search algorithm that attempts to minimize the user's utility, which is represented as a linear combination of planning time and plan cost. If plan cost is makespan, then the utility measures the 'goal achievement time', or the time from when the goal is presented to the planner, and planning starts, to when the plan finishes executing, and the goal is achieved by the agent. Bugsy is a best-first search algorithm, and relies on an estimate of remaining planning time similar to that of DAS in order to estimate the utility of each node it expands. While Bugsy is sensitive to its own planning time, it is not cognizant of external timed events such as deadlines, and does not prune nodes based on temporal information.

Related concepts in the search community include real-time search and anytime search. In the real-time search setting, the planner must return within a prespecified time bound the next action for the agent to take. This differs from our setting, in which the planner must return a complete plan and the temporal constraints are fine-grained and can relate individual domain propositions to absolute times. In anytime search, a planner quickly finds a complete plan, and then uses additional computation time to improve it until either it is terminated by an external signal or an optimal solution is found. In our setting, the planner may not run indefinitely, but rather is expected to minimize the agent's goal achievement time. And while doing so, we demand that the planner recognize that time is passing and that it be responsive to timed events in the external world.

Encoding Planning and Execution Time

Many temporal planners (e.g., (Coles et al. 2009; 2012; 2010; Benton, Coles, and Coles 2012; Fernández-González, Karpas, and Williams 2015; 2017)) rely on an internal Simple Temporal Network (STN) (Dechter, Meiri, and Pearl 1991) (or possibly a linear program or a convex optimization problem — but we will abuse terminology and call all of these the STN) to represent the temporal constraints between the set of the *time points* where actions start or end. Specifically, planners that support required concurrency (Cushing et al. 2007) tend to use this representation to support concurrent execution of actions.

When planning is done offline, the STN contains some time point t_{ES}, which is the start of plan execution, and is assigned the value of 0. For convenience, we split each occurrence of action a in the plan into two snap-actions: $a_{\vdash}$ and $a_{\dashv}$, corresponding to the start and end of the action, respectively. For each of these we have a corresponding time point in the STN: $t(a_{\vdash})$ when a starts, and $t(a_{\dashv})$ when a ends. Actions which have started but not yet finished will only have the start time point, since this is a partial plan (as noted earlier, all starts eventually need to be paired with an end, but this is not a requirement of plans that are still under construction). Temporal constraints between the time points are either action *duration* constraints (between the time points of the same action occurrence), or *sequencing* constraints due to causal relations between actions. For example, if the end of action a achieves the start conditions of action b, then we would have $t(a_{\dashv}) - t(b_{\vdash}) \geq \epsilon$, where ϵ is the minimum separation between two events that depend on each other (Fox and Long 2003). Or, if the start of c threatens the preconditions of d, then $t(c_{\vdash}) - t(d_{\dashv}) \geq \epsilon$. Additionally, timed initial literals (TIL) (Edelkamp and Hoffmann 2004) are encoded into the STN by adding a time point $t(f)$ for the occurrence of TIL f, with the temporal constraint $t(f) - t_{ES} = \mathit{time}(f)$, where $\mathit{time}(f)$ is the time at which f occurs, as specified in the problem definition. These are then ordered with respect to the other steps in the plan by, again, adding sequencing constraints due to the causal relations between $\mathit{lit}(f)$ and the other steps in the plan.

In this paper, we focus on *online* planning. We want to account for the fact that time passes *during* the planning process, and that, in fact, planning time and execution time are both the same. In order to do so, we modify the STN described above by adding two additional time points: t_{PS} which is the time when planning started, and t_{now} which is the current time. We add the temporal constraint that $t_{now} - t_{PS}$ equals the currently elapsed time in planning. The expression $t_{ES} - t_{now}$ corresponds to the remaining planning time, which is, of course, unknown. We will discuss this expression, and how to treat it, in the next section. Now, $t_{PS} = 0$, while t_{ES} is unknown. Finally, because TILs describe absolute time, we must modify the temporal constraints corresponding to TILs to use t_{PS} instead of t_{ES}, i.e., the temporal constraint for TIL l would be $t(l) - t_{PS} = \mathit{time}(l)$, where $\mathit{time}(l)$ is the time at which l must occur.

Time-Aware Planning

We have described a technique for encoding an STN which captures the fact that execution only starts after planning ends, and planning takes time. We now describe the impact this has on search within a temporal planner.

Forward Planning Search Space

We take as our basis the forward-search approach of the planner OPTIC (Benton, Coles, and Coles 2012). Here, each search state comprises the plan π (of snap actions) that reaches that state; the propositions $p \subseteq F$ that hold after π was executed from the initial state; and the Simple Temporal Network $STN(\pi)$ encoding the temporal constraints over π.

When expanding a state in OPTIC, successors were generated in one of three ways:

- By applying a *start* snap-action that is logically applicable: any $a_{\vdash}$ where $p \vDash \mathit{cond}_{\vdash}(a)$; $\mathit{eff}_{\vdash}(a)$ would not break the invariant condition of an action that has started in π but not yet ended; and $\mathit{cond}_{\leftrightarrow}$ would be satisfied once $a_{\vdash}$ has been applied. In this case, in the successor state, $\pi' = \pi + [a_{\vdash}]$, p is updated according to $\mathit{eff}_{\vdash}(a)$ to yield p', and a variable $t(a_{\vdash})$ added to $STN(\pi')$. Sequence constraints are imposed on this such that it follows any step in π that met one of $\mathit{cond}_{\vdash}(a)$; or whose effects refer to the same propositions as $\mathit{eff}_{\vdash}(a)$; or whose preconditions (including invariant conditions) would be threatened by $\mathit{eff}_{\vdash}(a)^1$.
- By applying an *end* snap-action that is logically applicable – any $a_{\dashv}$ where a has started in π but not yet ended; $p \vDash \mathit{cond}_{\dashv}(a)$; and whose effects $\mathit{eff}_{\vdash}(a)$ would not break the invariant of *any other* action that has started in π but not yet ended. In this case, the successor state is updated in a way analogous to starting an action, with the additional STN constraint $\mathit{dur}_{\min}(a) \leq t(a_{\dashv}) - t_{a_{\vdash}} \leq \mathit{dur}_{\max}(a)$.
- By applying a *Timed Initial Literal* l that has not already occurred in π. In this case, $\pi' = \pi + [l]$, p is updated according to $\mathit{lit}(l)$ to yield p', and a variable $t(l)$ is added to $STN(\pi')$. For the purposes of sequence constraints, this can be thought of as being a snap-action with no preconditions – it suffices to order it after steps in π whose preconditions or effects refer to $\mathit{lit}(l)$. To fix the time at which l occurs, an additional STN constraint is added: $t(l) - t_{PS} = \mathit{time}(l)$ – while snap-actions are ordered only relative to other points in the plan, TILs must also occur a specific amount of time after time zero.

State expansion in this way generates candidate successors that are logically feasible; to ensure they are also temporally feasible, only those whose STNs are consistent are kept. Using an all-pairs shortest path algorithm in the STN will both check consistency (with negative cycles corresponding to an inconsistent STN), and give us the earliest and latest possible time at which each snap-action could be applied. We denote these $t_{\min}(x)$ and $t_{\max}(x)$ for each STN variable $t(x)$. Typically, only the former of these is used – to map π to a schedule σ, each start–end snap-action pair $a_{\vdash}$, $a_{\dashv}$ gives a triple $\langle a, t_{\min}(a_{\vdash}), (t_{\min}(a_{\dashv}) - t_{\min}(a_{\vdash}))\rangle$. In other words, apply each action as soon as possible, with the shortest possible duration, thereby minimizing execution time.

Extending this approach to planning while aware of planning and execution time requires a number of modifications, which we now step through.

No action can start before plan execution starts – because execution cannot start until a plan has been produced. That is, for each $a_{\vdash}$ in the plan π, we add a constraint $t_{ES} \leq t(a_{\vdash})$ to the STN, where t_{ES} is the time at which execution will start. We do not know this *a priori*, but can at least say $t_{now} \leq t_{ES}$ is the time since the planner started executing. An STN for a plan produced during successor generation will then be consistent *iff* it is not already too late to start executing the plan.

These additional constraints can be thought of as pushing the earliest actions in the plan to start after now; the effects of which are then propagated through the STN to appropriately delay the later actions, according to the sequence and duration constraints. If an otherwise-consistent STN is made inconsistent by these, then necessarily there must be a snap-action x where $t_{\max}(x) < t_{now}$ – i.e. we are past the latest point at which x could have been applied.

[1] As search progresses in a strictly forward direction, all threats are dealt with by *demotion* – ordering the new step after existing steps.

Planning time particularly matters in the presence of TILs – in the absence of these, we can start executing a plan whenever we like by simply delaying the start of the first action. If TILs are present, though, these anchor the plan to having to fit around absolute time: with reference to state expansion, when a TIL is added to the plan, this fixes it to come after any earlier steps with which it would interfere, thereby constraining their maximum time.

Automatically applying past TILs – if we are now past the time at which a TIL has occurred, it is added to π before expanding the state.

More formally, immediately before expanding a state $S = \langle \pi, p, STN(\pi) \rangle$, the following TILs are applied:

$$\{l \in T \mid t(now) \geq time(l) \wedge l \notin \pi\}$$

If there are several such TILs, they are applied in ascending order of $time(l)$. The mechanism for applying these TILs is identical to that in OPTIC: each is applied, to yield a successor state S'; and then S' replaces S. By doing this before expanding the state, we account for time having passed since S having been placed on the open list, and it being expanded – if in this time a TIL will have happened, S is updated accordingly, before expansion.

If this modification was not made, search would be free to branch over what step should next be added to π. In the case where a TIL l represents a deadline – by deleting a precondition on actions that must occur by a given time – search would be free to apply these actions, even though in reality it is too late. By forcing the application of past TILs, we avoid this behavior: all such actions would then become inapplicable.

Pruning states where it is too late to start their plan From the STN for a plan π, we can note the latest point at which that plan can start executing; and prune any states for which this time has already passed.

As noted earlier, to check if the STN for a state is consistent, we use an all-pairs shortest path algorithm. This incidentally yields the minimum and maximum time-stamps for each snap-action. For snap-actions that are ordered before a TIL – which are fixed in time – these maximum time-stamps are finite. Moreover, because the plan is expanded in a strictly forward direction, the maximum timestamps are monotonically decreasing: it is not possible to somehow order a new action before a plan step, in a way that reduces its maximum time-stamp. Thus, for each state $S = \langle \pi, p, STN(\pi) \rangle$ we identify the start snap-action in π that has the earliest possible maximum time-stamp – this is the latest time at which π could feasibly be executed:

$$latest\_start(\pi) = \min\{t_{\max}(a_{\vdash}) \mid a_{\vdash} \in \pi\}$$

Then, when S is about to be expanded – after it was generated, placed on the open list, and then removed – it is pruned if $t_{now} > latest\_start(\pi)$.

Experiments

To gain a concrete sense of the practical import of our technique, we experimentally compared it to the baseline method of prespecified planning times. We performed experiments in two types of domains: a realistic AUV simulation, and a set of IPC domains.

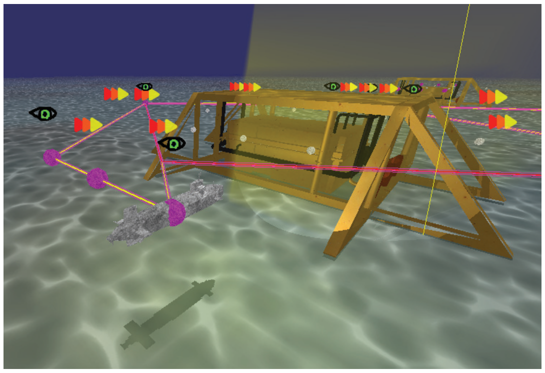


Figure 1: Screenshot of the underwater simulator, in which the AUV is inspecting the structure.

As a baseline against which to compare our time-aware planner, we used OPTIC in optimization mode, searching for the best plan within a varying fixed planning time of T seconds. Time windows were considered to be T seconds earlier, to adjust the initial state to the start of execution time. Therefore, a TIL l occurring at time $time(l)$ seconds, using a planning time of T seconds, will occur at time $(time(l) - T)$ (at least 0) in the plan.

AUVs

We demonstrate the approach in simulation with autonomous underwater vehicles (AUVs). We embed OPTIC and our planner into ROS, using ROSPlan (Cashmore et al. 2015), to control the AUV. The AUV is equipped with a manipulator and placed in an underwater structure, with the task to inspect certain areas and to ensure that valves are turned to correct angles. The valves can only be turned within certain time windows, outside of which the valve is blocked. If the valve cannot be turned to the correct angle within an early time window, then a later window can be used. We generated 41 missions with varying time windows. A screenshot of the simulation is shown in Figure 1

These missions normally form part of a larger, strategic mission, spread out over a number of seabed manifolds. The AUV moves between these manifolds in order to complete the missions. Due to the uncertainty in the environment, it is not known beforehand precisely what time the AUV will arrive at the manifold. Before beginning the task, the AUV must construct a new plan. Plans with shorter durations are considered to be of higher quality, as this eases the time constraints on the remainder of the missions. We use this scenario to show that our approach allows the AUV to make use of earlier time windows, generating plans of higher quality.

The results are summarized in Table 1. The table shows the number of problems solved for each planner, out of a possible 41. Using our approach every problem was solved. Using a fixed planning time, some problems were unsolvable due to a planning time that was too short. The table

	Time Aware	OPTIC$_{50}$	OPTIC$_{100}$	OPTIC$_{200}$
best quality	34	13	20	19
IPC quality	40.19	25.55	26.19	26.47
problems solved	41	26	34	40

Table 1: Table comparing the number of problems solved, the number of plans of highest quality, and the IPC quality for each approach.

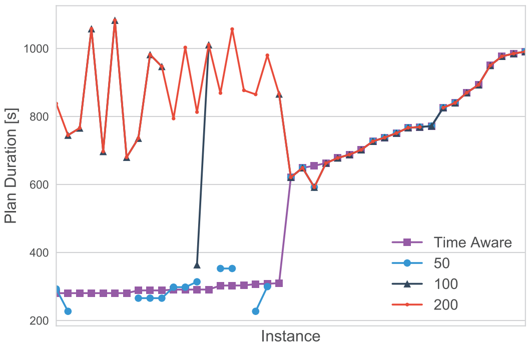


Figure 2: Plan durations per problem for each approach. The time-aware approach solves many problems using an earlier time window. OPTIC using a long planning time solves almost every problem, but only using the later time windows. Other planning time bounds are less reliable.

also shows the number of best plans for each approach. This is the number of problems for which that approach produced the plan of highest quality between the four approaches (possibly jointly). There it can be seen that although increasing the planning time allows for all problems to be solved, the quality is much poorer. The higher absolute number of best plans for the 200 second planning time is due to the greater number of problems solved. Finally, the table shows the IPC quality, calculated for all problems. These results demonstrate the choice between acting quickly, utilizing early time-windows, or producing plans reliably. Using the time-aware approach does both.

This can be seen more clearly in Figure 2. This figure compares the plan duration from each approach per problem. Using OPTIC$_{200}$ almost every problem is solved, at the longest possible plan duration – assuming planning takes 200 seconds forces the planner to have to use the later time windows. Other approaches may generate shorter plan durations, but fail to solve many of the problems.

IPC Domains

In our IPC experiments, we tested all IPC-4 and IPC-5 domains that contain TILs: airport, pipesworld, satellite, truck, and UMTS. The UMTS domains and half of the airport instances were omitted as none of the planners completed these. The planners were given a maximum of $200s$ of CPU time and 4GB of memory.

	Time Aware	OPTIC$_{0.1}$	OPTIC$_{1}$	OPTIC$_{10}$
best quality	38	1	0	1
IPC quality	38	9.99	29.74	19.89
problems solved	38	10	30	21

Table 2: Table comparing the number of problems solved, the number of plans of highest quality, and the IPC quality for each approach

Table 2 presents results on the modified IPC domains. The fixed planning time planners were outperformed by the time-aware methods in every domain. Several instances were unsolvable by the former due to the fixed planning time constraints. Table 3 shows the planners detailed performance in each relevant domain tested.

In addition to the fixed planning times that are showed in Table 2 and Table 3 we have tested $50s$, $100s$, and $200s$. The performance of the baseline approach with these planning times were lower than the time-aware method and the best presented baseline, thus these results were omitted.

Conclusions and Future Work

We have presented a domain-independent temporal planner that takes the interaction between the time spent on planning and execution time into consideration. We have demonstrated empirically that this planner achieves much better results in domains with absolute deadlines than our baseline approach. However, our work is merely the first step in addressing this important topic. There remain many exciting avenues for future work.

For example, our planner only looks at the current partial plan, and uses a heuristic to "look" into the future. This heuristic is used to estimate the remaining search depth, but not to obtain more information about future actions and their effects on deadlines. In order to get a more informed view of future actions, and their effect on deadlines, we will explore using temporal landmarks (Karpas et al. 2015). These landmarks could be encoded into the same STN of the partial plan, and thus we believe we will be able to achieve even better pruning of branches of the search tree which will not lead to a solution in time.

More broadly, the problem we are addressing here could benefit from more explicit metareasoning (Russell and Wefald 1991). For example, suppose we had a planning problem with two possible solutions, each of which must be explored on a separate branch of the search tree. Further suppose that each of these solutions has a deadline which leaves just enough time to explore one of the branches, but not both of them. Clearly, a planner with perfect knowledge would choose one of these branches and explore it. On the other hand, the approach we present here will explore both branches until it realizes there is not enough time left, and will then prune both branches — without solving the problem. In future work, we will explore ways of addressing this type of problem by incorporating explicit metareasoning on

group	planner	solved	time	GAT
airport-1	**Time Aware**	**14**	6.62	193.54
	OPTIC$_{0.1}$	2	0.06	89.61
	OPTIC$_{1}$	10	0.24	167.72
	OPTIC$_{10}$	10	0.20	176.72
pipesworld	Time Aware	3	0.72	16.06
	OPTIC$_{0.1}$	1	0.05	12.11
	OPTIC$_{1}$	**4**	0.51	15.51
	OPTIC$_{10}$	0		
satellite-1	**Time Aware**	1	0.03	190.23
	OPTIC$_{0.1}$	1	0.04	190.31
	OPTIC$_{10}$	1	0.02	200.21
	OPTIC$_{1}$	1	0.02	191.21
satellite-2	**Time Aware**	**5**	0.71	181.89
	OPTIC$_{0.1}$	1	0.03	190.31
	OPTIC$_{1}$	4	0.39	182.87
	OPTIC$_{10}$	1	0.56	129.16
satellite-3	**Time Aware**	**5**	0.80	181.88
	OPTIC$_{0.1}$	1	0.03	190.31
	OPTIC$_{1}$	4	0.36	182.87
	OPTIC$_{10}$	1	0.56	129.16
satellite-4	**Time Aware**	**4**	2.20	165.20
	OPTIC$_{0.1}$	0		
	OPTIC$_{1}$	2	0.15	155.00
	OPTIC$_{10}$	2	1.38	147.38
truck	**Time Aware**	**6**	0.21	1840.98
	OPTIC$_{0.1}$	4	0.05	1673.95
	OPTIC$_{1}$	5	0.06	1674.20
	OPTIC$_{10}$	**6**	0.20	1855.97

Table 3: Table comparing the number of problems solved, the planning time, and the goal achievement time (GAT) grouped by IPC instance type. The planning time, and the GAT is the mean of all instances in the group solved by the planner.

planning time allocation into the search strategy.

One possible approach for this would be to treat the expression $t_{ES} - t_{now}$ as a variable, which we will denote by *slack*. We can then treat the STN as a mathematical optimization problem, and maximize the slack. The slack for node n can serve as a proxy for the probability of finding a solution in time in the subtree rooted at n. Our metareasoning algorithm could then choose the next node to expand based on both heuristic estimates and the slack.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 Research and Innovation programme under Grant Agreement No. 730086 (ERGO).

References

Benton, J.; Coles, A. J.; and Coles, A. 2012. Temporal planning with preferences and time-dependent continuous costs. In *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS).*

Burns, E.; Ruml, W.; and Do, M. B. 2013. Heuristic search when time matters. *Journal of Artificial Intelligence Research* 47:697–740.

Cashmore, M.; Fox, M.; Long, D.; Magazzeni, D.; Ridder, B.; Carrera, A.; Palomeras, N.; Hurtós, N.; and Carreras, M. 2015. Rosplan: Planning in the robot operating system. In *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS)*, 333–341.

Coles, A.; Fox, M.; Halsey, K.; Long, D.; and Smith, A. 2009. Managing concurrency in temporal planning using planner-scheduler interaction. *Artificial Intelligence* 173(1):1–44.

Coles, A. J.; Coles, A.; Fox, M.; and Long, D. 2010. Forward-chaining partial-order planning. In *Proceedings of the 20th International Conference on Automated Planning and Scheduling (ICAPS)*, 42–49.

Coles, A. J.; Coles, A.; Fox, M.; and Long, D. 2012. COLIN: planning with continuous linear numeric change. *Journal of Artificial Intelligence Research (JAIR)* 44:1–96.

Cresswell, S., and Coddington, A. 2003. Planning with timed literals and deadlines. In *Proceedings of 22nd Workshop of the UK Planning and Scheduling Special Interest Group*, 23–35.

Cushing, W.; Kambhampati, S.; Mausam; and Weld, D. S. 2007. When is temporal planning really temporal? In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 1852–1859.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49(1-3):61–95.

Dionne, A. J.; Thayer, J. T.; and Ruml, W. 2011. Deadline-aware search using on-line measures of behavior. In *Proceedings of the Symposium on Combinatorial Search (SoCS-11).*

Edelkamp, S., and Hoffmann, J. 2004. PDDL2.2: The language for the classical part of the 4th international planning competition. Technical Report 195, University of Freiburg.

Fernández-González, E.; Karpas, E.; and Williams, B. C. 2015. Mixed discrete-continuous heuristic generative planning based on flow tubes. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, 1565–1572.

Fernández-González, E.; Karpas, E.; and Williams, B. C. 2017. Mixed discrete-continuous planning with convex optimization. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 4574–4580.

Fikes, R. E., and Nilsson, N. J. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. In *Proceedings of the 2nd International Joint Conference on Artificial Intelligence (IJCAI)*, 608–620.

Fox, M., and Long, D. 2003. PDDL2.1: an extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research (JAIR)* 20:61–124.

Karpas, E.; Wang, D.; Williams, B. C.; and Haslum, P. 2015. Temporal landmarks: What must happen, and when. In *Pro-*

ceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS), 138–146.

Niemueller, T.; Lakemeyer, G.; and Ferrein, A. 2015. The RoboCup Logistics League as a Benchmark for Planning in Robotics. In *WS on Planning and Robotics (PlanRob) at Int. Conf. on Aut. Planning and Scheduling (ICAPS)*.

Ruml, W.; Do, M. B.; Zhou, R.; and Fromherz, M. P. J. 2011. On-line planning and scheduling: An application to controlling modular printers. *Journal of Artificial Intelligence Research* 40:415–468.

Russell, S. J., and Wefald, E. 1991. Principles of metareasoning. *Artificial Intelligence* 49(1-3):361–395.

Creating and Using Tools in a Hybrid Cognitive Architecture

Dongkyu Choi
Department of Aerospace Engineering
University of Kansas
Lawrence, KS 66045 USA
dongkyuc@ku.edu

Pat Langley, Son Thanh To
Institute for the Study of Learning and Expertise
2164 Staunton Court, Palo Alto, CA 94306 USA
patrick.w.langley@gmail.com
son.to@knexusresearch.com

Abstract

People regularly use objects in the environment as tools to achieve their goals. In this paper we report extensions to the ICARUS cognitive architecture that let it create and use combinations of objects in this manner. These extensions include the ability to represent virtual objects composed of simpler ones and to reason about their quantitative features. They also include revised modules for planning and execution that operate over this hybrid representation, taking into account both relational structures and numeric attributes. We demonstrate the extended architecture's behavior on a number of tasks that involve tool construction and use, after which we discuss related research and plans for future work.

Introduction

The ability to create and use complex tools is a distinctive feature of human cognition. People use objects in their surroundings to help achieve goals, sometimes combining multiple objects into a new tool that fits their need. This involves planning but focuses on constructing physical artifacts to achieve other ends, rather than generating isolated action sequences. For example, scenes from a popular television series, MacGyver, often depicts the protagonist creating tools from materials that seem unrelated to his objectives. The character ingeniously uses objects in ways for which they were not intended, often combining them into a tool for his purpose. Current AI systems, including our current work, do not demonstrate such creative abilities.

In this paper, we report progress toward intelligent agents that exhibit the ability to create and use physical tools. Our approach extends an existing cognitive architecture to support this capacity. In the next section, we present a scenario that illustrates how tool construction and use can help an agent achieve its goals. Next we briefly review ICARUS, an architecture for physical agents, and we describe extensions to its representation and processes that let it create and use tools. After this, we report runs in a simulated environment, drawing on the scenario presented earlier, that demonstrate the revised system's abilities. We conclude by discussing related work and plans for future research.

A Motivating Scenario

We can clarify the challenge of tool creation and use with a scenario. Consider a robot that wants to escape from inside a crumbled building. Its goal is to move from a location inside the building to another one outside, but between them is a wide gap in the floor that the robot cannot traverse and an opening in the wall that is too high for it to reach without other support. The robot observes some wooden planks of different lengths and thicknesses. Knowing its own weight and the maximum height it can climb, it stacks planks across the gap to build a bridge that will support its weight. The robot then crosses the bridge and thus traverses the gap. In a similar fashion, it builds a staircase to the opening on the wall, goes up the staircase, and escapes from the building.

In this scenario, the robot manipulates objects in its environment and assembles them into tools which it then uses to achieve its goal. To create the right tool, it considers both structural and numeric factors. Wooden planks laid over the gap can serve as a basic bridge, but they must be long enough to cross the gap and strong enough to hold the robot's weight. A single plank may appear qualitatively sufficient, but a second plank may be needed to make the bridge strong enough. For an effective staircase, the building blocks must be arranged to give enough footing on each step and the steps should be no higher than the robot can climb.

We can view bridges and staircases as tools that are constructed from available components. Computing the load a bridge must hold or the height of a step requires quantitative reasoning about objects' positions and dimensions, but the agent must first devise some qualitative structure that its numbers describe. We believe the scenario provides a reasonable challenge for testing an intelligent agents' ability to create and use tools, as it requires a combination of qualitative and quantitative reasoning.

A Brief Review of ICARUS

ICARUS (Langley, Choi, and Rogers 2009) is a cognitive architecture that provides an infrastructure for building intelligent agents that operate in physical settings, simulated or actual. As with other architectures like Soar (Laird et al. 1986) and ACT-R (Anderson and Lebiere 1998), it makes commitments about the representation of content, the memories that store that information, and the processes that manipulate it. ICARUS incorporates many ideas from cognitive psy-

Table 1: Sample ICARUS concepts for the staircase problem.

```
((on ?o1 ?o2)
 :elements ((block ?o1 ^x ?x1 ^y ?y1 ^length ?length1)
            (block ?o2 ^x ?x2 ^y ?y2 ^length ?length2
                       ^height ?height2))
 :tests    ((*overlapping ?x1 ?length1 ?x2 ?length2)
            (= ?y1 (+ ?y2 ?height2))))

((staircase ?o ?o1)
 :elements   ((block ?o ^height ?h))
 :conditions ((on ?o ?o1)
              (staircase ?o1 ?o2)
              (step-size ?step))
 :tests      ((<= ?h ?step)))
```

Table 2: Sample ICARUS skills for the bridge problem.

```
((pick-up ?o)
 :elements   ((robot ?robot)
              (block ?o))
 :conditions ((clear ?o) (not (holding ?robot ?any)))
 :actions    ((*pick-up ?robot ?o)))
 :effects    ((holding ?robot ?o))

((build-bridge ?block ?bottom)
 :elements   ((block ?block))
 :conditions ((bridge ?top ?bottom))
 :subskills  ((stack ?block ?top))
 :effects    ((bridge ?block ?bottom))
```

chology, but it emphasizes construction of intelligent systems that carry out complex activities rather than fitting the results of psychological experiments. In this section, we review the architecture, starting with assumptions for representation and memories and then describing its mechanisms for inference, reactive execution, and problem solving.

Representation and Memories

ICARUS distinguishes between two forms of long-term knowledge: *concepts* that underlie inference and procedural *skills* that support activity. The framework also separates *percepts* from the environment from *beliefs* inferred about them. The former describe observed objects in terms of their attributes, typically numeric, while the latter take the form of relational literals like *(on A B)*. This distinction will figure centrally later in the paper. The conceptual knowledge base links percepts to beliefs through a set of defined *concepts*. Each conceptual rule specifies the conditions that must match to infer a belief of a given type. The conditions of a *primitive* concept refer only to percepts and their attribute values, whereas the conditions of a *nonprimitive* concept also refer to more basic conceptual predicates.

Table 1 shows some ICARUS concepts that describe relations and situations for the staircase scenario. The first conceptual rule, for the predicate *on*, is primitive, as it has only an *:elements* field, which describes perceived objects and their attributes, along with a *:tests* field that constrains the matched variables. This concept refers to two *block* objects and checks numeric relations between their positions, lengths, and heights. The second concept, for the predicate *staircase*, is nonprimitive, as it refers to other concepts in its *:conditions* field. These include the concepts like *on*, *step-size*, and *staircase*, so the definition is recursive. Thus, concepts are organized into a hierarchy, with more complex predicates defined in terms of simpler ones.

ICARUS skills build on its conceptual knowledge. Each skill clause includes generalized percepts, conditions that must hold for application, and effects that its application produces. A *primitive* skill clause refers to some action that the agent can execute directly in the environment, whereas a *nonprimitive* skill clause refers to other, more basic, skills.

Table 2 shows examples of ICARUS skills relevant to the bridge problem in our scenario. The first skill clause, *pick-up*, refers to two perceived objects, a *robot* and a *block*, and has two conditions, one positive (for *clear*) and the other negative (for *holding*). This clause is primitive because it includes the executable action **pick-up*. The second skill, *build-bridge*, mentions one percept and one conceptual condition, but it is nonprimitive because it includes the subskill *stack*. Such references organize skills into a hierarchy in which primitive clauses serve as terminal nodes, much as in a hierarchical task network (e.g., Nau et al. 2003).

Cognitive Processes in ICARUS

The architecture utilizes its concepts and skills during processing, which operates in four-step cycles. First, ICARUS deposits percepts from the environment in a perceptual buffer. The system does not model the extraction of percepts from sensors, but they serve as plausible outputs of sensory processing. Second, the architecture combines its conceptual knowledge with these percepts to infer beliefs that hold for the current situation. ICARUS matches primitive conceptual clauses against perceived objects to generate low-level beliefs, then matches nonprimitive concepts against them to produce higher-level beliefs. For example, the first clause in Table 1 generates a belief about the *on* relation when a block's *y* position equals that of another block plus its height and when the **overlapping* test is true.

Once ICARUS has inferred beliefs about the current situation, an execution stage attempts to find a path downward through the skill hierarchy that it can carry out in the environment. This module starts with a top-level goal, retrieves a skill clause that should achieve it and has conditions satisfied by current beliefs. If this skill instance is primitive, the architecture executes its associated action; if not, then it considers matched subskills. This recursive process returns a path through the skill hierarchy whose execution should bring the agent closer to its goal(s). When ICARUS cannot find such an applicable path, it invokes a problem-solving module that carries out search for sequences of skills which achieve the current goals. Execution and problem solving are tightly interleaved, with the system carrying out selected skill instances when applicable and resorting to problem solving when it encounters an impasse.

We should note that, although ICARUS grounds its concepts and skills in quantitative percepts and actions, the in-

Table 3: An ICARUS concept that illustrates the extended numeric representation.

```
((bridge ?b ?g ?leftend ?rightend)
 :elements   ((block ?b ^x ?leftend ^length ?ln)
              (gap ?g ^left ?gl ^right ?gr))
 :attributes (?left is (- ?gl 1)
              ?right is (+ ?gr 1)
              ?rightend is (+ ?leftend ?ln))
 :tests      ((<= ?leftend ?left)
              (>= ?rightend ?right)))
```

ference, execution, and problem-solving modules primarily produce qualitative and relational structures. This does not keep the architecture from operating in continuous domains like simulated urban driving (Langley et al. 2009; Choi 2011), but we will see that it raises challenges for the construction and use of complex tools.

Numeric Representation and Processing

As noted earlier, reasoning about tools often requires that an agent operate over not only qualitative aspects of the environment, but also its quantitative properties. In this section, we discuss two extensions to ICARUS that let the architecture support numeric processing, the first involving representation and the second concerning planning.

Representational Extensions

ICARUS receives and processes perceptual elements that include types, names, and attribute-value pairs for objects in the world. The original system can represent symbolic relations among objects and concepts can include simple tests on numeric attributes. But it cannot reason about numeric relations or specify arithmetic computations and associate their results with a new variable. In previous research, this limitation has caused problems when using ICARUS to control physical robots, where the continuous domain requires encoding of numeric constraints. Naturally, this issue also arises in tool creation and use. To address the problem, we extended the conceptual formalism to specify arithmetic combinations of numeric attributes and associate them with new variables that can appear elsewhere in the concept.

Table 3 shows a sample concept that uses this extended notation. The clause includes a new field, *:attributes*, that specifies desired numeric calculations and their variable assignments. This specific clause states that the position of a block's right side (denoted by the variable *?rightend*) can be computed from its left side position, *?leftend*, and its length, *?ln*. The concept also specifies how to compute the left and right positions, *?left* and *?right*, for a spatial gap with one unit margins at both ends. These values are also used, along with the left and right ends, in two inequality tests.

This extension lets ICARUS specify numeric calculations and how to reuse their results elsewhere in a conceptual clause, complementing the qualitative structures it could already express. However, this only describes the environmental situation, not how agent's actions will alter it. In response, we also extended the notation for skills to incorporate details about quantitative effects of their execution.

Table 4: An ICARUS skill for creating a bridge that illustrates the extended numeric formalism.

```
((fill-gap-center ?b ?g)
 :elements   ((block ?b ^x ?x0 ^length ?l ^weight ?w)
              (robot ?robot ^weight ?weight
                            ^status ?status ^holding ?b)
              (gap ?g ^left ?gl ^right ?gr))
 :actions    ((*fill-gap-center ?robot ?b ?gl ?gr))
 :effects    ((bridge ?b ?g ?x0 (+ ?x0 ?l))
              (block ?b ^x (/ (- (+ ?gl ?gr) ?l) 2) ^y 0
                        ^len ?l ^weight ?w)
              (robot ?robot ^weight (- ?weight ?w)
                     ^status ?status ^holding nothing)))
```

Table 4 shows a skill that takes advantage of this extension. The main change is in the *:effects* field, which describes the outcome of a skill's successful execution. Previously, this field could only include symbolic effects about relational beliefs that would become true or false after application. In the new notation, the field can describe expected changes not only in symbol structures, but also in the numeric attributes of objects. The skill will not only cause the symbolic relation *(bridge . . .)* to become true, but also change the block's *x* position to the value of the expression, *(/ (- (+ ?gl ?gr) ?l) 2)*, and reduce the robot's *weight* by *?w*.

Extensions to Processing

The original architecture could match against numeric attributes of perceived objects, but it could neither perform mathematical calculations over these numbers nor allow the results in concept heads. The representational changes to concepts and skills remedies these limitations, but taking advantage of them also required us to augment ICARUS's information processing along two fronts. The first deals with inference, which now calculates the values of arithmetic expressions in concepts and binds them to specified variables that may appear in the heads. These numeric values, in turn, can influence inference of symbolic beliefs at higher levels, as they are carried upward through the hierarchy during the conceptual inference process.

These changes to the formalism require no alteration of the execution module, but they do necessitate changes to problem solving. In response, the revised module computes not only symbolic relations during its mental execution of skills but also numeric values associated with them. The new problem solver utilizes forward chaining, which lets the system update numeric attributes of an object, add new literals, or delete existing literals from the state using information encoded in skills' *:effects* field. Such mental execution has direct effects on the projected state, but indirect changes can also occur, which the architecture determines by invoking the inference module. As a result, the problem solver can generate plans that satisfy both symbolic and numeric requirements specified in the agent's goals.

Encoding and Processing Virtual Objects

Despite its new ability to reason about quantitative aspects of the environment, the extended ICARUS still cannot recognize an existing object as a potential tool or reason about how to create one from available elements. This is because the architecture only recognizes primitive objects as distinct entities, not combinations of them. In contrast, people readily view composite structures as objects themselves, describe numeric features associated with them, and reason about them as unified entities. To create and utilize complex tools, ICARUS needs the ability to reify and process such *virtual* objects in its environment.

Representational Extensions

The ability to include numeric attributes in concept heads paves the way to handling virtual objects. Without this extension, the architecture can infer beliefs only as symbolic literals, which makes them different from perceived objects in that they lack numeric attributes. Previously, for example, a *bridge* concept that describes a composite object could only produce a symbolic belief that informs the agent about its existence. In contrast, the new version can calculate the values for numeric attribute associated with the *bridge* entity, such as its thickness and weight limit.

However, computing such numeric attributes is not enough. We also need some way to associate them with the virtual object, which requires giving it a symbolic identifier in the same manner as percepts. This extension effectively eliminates the distinction in the original ICARUS between beliefs and percepts, so the new architecture stores them in a single working memory. The only remaining differences are that percepts come directly from an external environment, while beliefs are inferred, and that beliefs include a symbolic relation, while percepts lack them. Of course, we can apply this idea recursively to specify higher-level virtual objects in terms of lower-level ones.

For example, the two conceptual clauses for *bridge* that appear in Table 5 not only describe the class of *situations* in which one or more blocks cover a gap, but also specify a new *virtual object* that denotes the bridge. This composite object has its own attributes, such as its left position, right position, and weight, the values of which are calculated from the attribute values of its component objects.

Implications for Processing

Once the extended ICARUS has created virtual objects, it can use them as if they were objects perceived directly in the environment. The second, recursive, clause for *bridge* concept shown in Table 5 lets the system recognize situations in which a block is stacked on a bridge and generate another virtual object that is also a *bridge*, but one with a higher weight limit than the original one.

As the table shows, the new notation also changes the syntax for the `:elements` field. Here the expression *A is B* states that one should associate an identifier *A* with *B*, which may be a percept or a relational belief. Recall that percepts enter the perceptual buffer with such identifiers, but that ICARUS must name its beliefs before it can associate numeric attributes with them. The extended architecture retains the identifiers for these virtual objects in working memory, so they can appear as arguments in higher-level beliefs that result from conceptual inference.

Table 5: Some ICARUS concepts that specify virtual objects.

```
((bridge ?b ^gap ?gl ^left ?l ^top-left ?tl
           ^top-right ?tr ^right ?r ^weight ?weight)
 :elements (?b is (block ?b ^x ?tl ^y 0 ^len ?len
                         ^weight ?weight)
           ?gl is (gap ?gl ?gr))
 :tests    ((<= ?tl (- ?gl 1))
           (>= (+ ?tl ?len) (+ ?gr 1)))
 :attributes (?l is ?tl
              ?tr is (+ ?tl ?len)
              ?r is (+ ?tl ?len)))

((bridge ?b ^gap ?gl ^left ?l ^top-left ?tl
           ^top-right ?tr ^right ?r ^weight ?weight)
 :elements (?b is (block ?b ^x ?tl ^y ?y ^len ?len
                         ^weight ?w)
           ?b1 is (bridge ?b1 ^gap ?gl ^left ?l
                      ^top-left ?tl1 ^top-right ?tr1
                      ^right ?r ^weight ?w1)
           ?b1 is (block ?b1 ^x ?tl1 ^y ?y1
                      ^len ?len1 ^weight ?w2)
           ?gl is (gap ?gl ?gr))
 :tests    ((<= ?tl (- ?gl 1))
           (>= (+ ?tl ?len) (+ ?gr 1))
           (= (+ ?y1 1) ?y)
           (<= (+ ?tl1 1) ?tl)
           (<= (+ ?tl ?len 1) ?tr1))
 :attributes (?weight is (+ ?w ?w1)
              ?tr is (+ ?tl ?len)))
```

What we have described suffices for ICARUS to draw inferences about composite objects, but not to use them for driving agent activity. Of course, virtual objects can also appear in the *effects* field of skills, which means that the problem solver can form expectations about their creation or destruction upon execution. This means, for example, that the agent can use its hierarchical skills to form plans that involve constructing composite objects which enable later steps that achieve its goals. But it can also use search to generate plans entirely from primitive skills and, by invoking the inference process, deduce that an action sequence has the side effect of creating a complex virtual object that it can use as a tool.

Demonstrations of the Extended Architecture

To confirm that the extended system behaves as intended, we carried out demonstration runs on the scenario described earlier. Here an ICARUS agent controls a simulated mobile robot to reach its destination. In one case, there is a chasm between the initial and the goal location; in another problem, the goal is at a higher location than the robot can reach directly. In both cases, the agent can use blocks of different sizes to build a bridge or staircase, which it can then use.

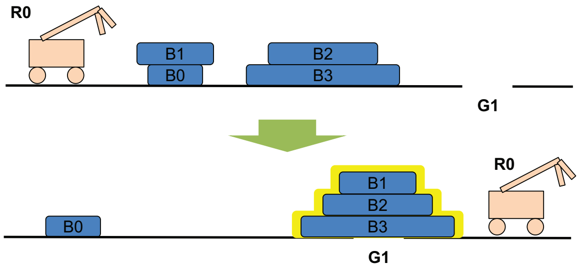


Figure 1: Initial and final states for one version of the bridge problem. The robot, *R0*, starts on the lefthand side and must use blocks to build a bridge over the gap, *G1*, to reach its goal on the righthand side.

Simplifying Assumptions

The primary aim of these demonstration runs was to show that the extensions to ICARUS, described earlier, support the creation and use of tools. For this reason, we introduced four simplifying assumptions that made the planning and execution tasks somewhat easier than they would be in a realistic simulation:

- Although ICARUS allows durative skills that require repeated application to achieve their effects, in the runs all skills produce results in one step;
- The 2D simulated environments let agents pick up and stack objects without first needing to approach them or to move around obstacles;
- Agents must use planning to find a sequence of skills that construct composite objects that can serve as tools, but skills for using them operate in one step; and
- We provided agents with hierachical concepts for tools that appear as conditions on these tool-using skills, effectively serving as affordances (Zech et al. 2017).

Ideally, future demonstrations should use more realistic simlated environments that eliminate these assumptions. Nevertheless, the reported runs offer clear proof of concept that the extended architecture can represent, reason about, construct, and use tools to achieve goals in continuous settings.

Creating and Traversing a Bridge

In the first setting, the robot must build a bridge to cross the chasm, using long wooden blocks of different lengths and strengths. The agent knows that, for the robot to traverse the bridge safely, it must: (1) cover the chasm by a margin of at least a foot at each end; (2) withstand the robot's weight and any payloads; and (3) if it is made from stacked blocks, include a staircase at each end with steps no higher than a foot and at least a foot wide. The agent has no skill that directly creates a bridge, so it must use problem solving to find some plan to build one that satisfies these requirements. The system must then execute this plan, building the bridge in the environment and crossing the chasm to reach its destination.

For this problem, we gave ICARUS four concepts and four primitive skills, including the ones shown in Tables 4 and 5. Using this knowledge, the agent can recognize situations in which a block is stacked on another, detect a bridge composed of blocks, pick up a block to either stack it on another or cover a chasm, and finally cross the bridge when it is complete. Figure 1 shows an initial state in which the robot perceives itself, a chasm, and four blocks that are two, four, six, and eight units in length and that have weight limits of one, five, one, and two, respectively. Block *B1* is on block *B0* and block *B2* is on block *B3*.

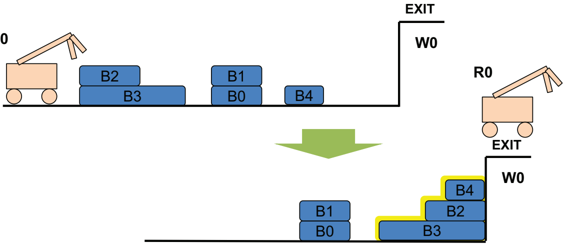


Figure 2: Initial and final states for one version of the staircase construction and climbing problem.

Given these initial and goal states, the problem solver uses forward-chaining search to find a plan that achieve its goal in nine steps. During this process, ICARUS first considers a bridge that only withstands a weight of two units, which is insufficient for the robot to cross. Next the system considers stacking a second block on the first to create a bridge with the maximum load of three units. This is still not sufficient, so it stacks yet another block, making a bridge that is strong enough for it to cross the chasm safely.

Once it has found this plan, the ICARUS agent executes it in the simulated environment over 29 cycles, first picking up *B2* to clear *B3* and stacking *B2* on *B1*. Next the system picks up the longest block *B3* and covers the gap with it. Then the robot picks up another block, *B2*, and stacks it on *B3* to create a stronger bridge, after which it stacks *B1* on the result to make it even stronger. At this point, the robot traverses the reinforced bridge to reach its goal.

We ran the extended architecture on 20 similar problems that involved four blocks of random lengths and weight limits. The system executed plans that had the average duration of 29.6 cycles with a standard deviation of 10.7 cycles. We also ran it, with the same knowledge, on a slightly different goal description in which the robot must carry a certain block as its payload across the chasm. In this altered scenario, the ICARUS agent generated a similar plan, this time requiring that it construct an even stronger bridge, then pick up the payload for delivery. Again, the robot executed this plan in the simulated world to achieve its goal.

Constructing and Climbing a Staircase

In the second scenario, the robot must escape from a room in which the exit is higher on the wall than it can reach without assistance. The environment contains long wooden blocks of different lengths that the agent can use to build a staircase for reaching the exit. The system knows that a staircase must: (1) have steps that are no taller than a foot for the robot to climb successfully and at least a foot wide so it can step on them safely; (2) be no further than a foot from the wall at its highest point; and (3) have a height that is within a foot of the exit's height. The robot must build a staircase that satisfies all these requirements before it can ascend and exit the room.

For this problem, we provided ICARUS with seven concepts and four primitive skills. The robot could use this knowledge to recognize situations in which one block is on top of another, categorize a virtual object as a staircase, pick up a block to either stack it on another or place it on the ground, and leave the room when it reaches the exit. Figure 2 shows one example of this scenario in which the robot perceives itself, the wall, and five blocks with lengths of 1.5, 1.5, 3, 4.5, and 1, respectively, and with heights of one unit.

The problem solver uses forward search to generate a plan that, in 13 steps, achieves the exit goal. During planning, ICARUS mentally constructs a staircase from three blocks that will let it leave the room, but only after considering shorter stairways. Once it has found this plan, the robotic agent executes it in the simulated environment, which takes 41 cognitive cycles. This involves picking up block *B4* to clear the area around the wall and stacking it on block *B1*. The agent then picks up block *B2* to clear *B3* and stacks *B2* on *B4*. The robot continues stacking the blocks *B3*, *B2*, and *B4*, in that order. At this point, it recognizes that it has built an acceptable staircase, so the robot climbs the stairs and exits the room, achieving its goal.

As another demonstration run, we used a variation on this problem that required the system to combine a number of shorter blocks to form steps for the staircase. This involved generating a more complex plan with additional steps that led to more virtual objects, greater search during planning, and longer execution times than in the first run, but the system handled them without any special difficulty.

In summary, the runs have demonstrated that the extended architecture can represent and reason about numeric attributes and virtual objects during inference, problem solving, and execution. This lets the revised ICARUS infer beliefs that incorporate numeric attributes, associate them with composite entities that its actions produce, and use this content to generate and carry out plans that achieve symbolic goals subject to numeric constraints. Together, these abilities support the construction of tools, such as bridges and staircases, from available components and their use once built.

Related Research

The extensions to ICARUS that let it create and use tools have clear precedents that merit discussion. We focus here on two contributions that we consider most important – reasoning over numeric attributes and using virtual objects. We have discussed the architecture's forward-chaining planning module elsewhere (To et al. 2015). We will not repeat our observations here except to note that it can use primitive skills, hierarchical ones, and their combination to generate plans, although the first option requires more search.

Research in cognitive architectures (Langley, Laird, and Rogers 2009) has emphasized symbolic representation and processing, due to their focus on high-level cognitive tasks. Nevertheless, well-established frameworks like Soar and ACT-R adopt an attribute-value notation that can easily encode the types of numeric object-based inputs we assume in both working memory and production rules. Both architectures have been used to control robotic agents, which certainly requires quantitative processing. However, they treat numeric manipulation as a special case of symbol processing, rather than giving them equal status, at the architecture level, as does the extended version of ICARUS.

Other paradigms also support a combination of symbolic and numeric processing. For example, logic programming emphasizes symbolic notations but can incorporate quantitative values and constraints, although they do not typically operate over time, as do ICARUS agents. AI planning systems also focus on symbolic tasks but have been adapted to include numeric content (e.g., Coles et al. 2012). These describe activity over time, but work in this tradition seldom supports the storage and use of hierarchical skills. Most robotic systems emphasize low-level numeric processing to the exclusion on high-level cognition. Hybrids like the 3T architecture (Bonasso et al. 1997) support both, but they adopt separate, specialized notations rather than offering a unified framework for cognition and action. Perhaps the closest robotics work (Levihn and Stilman 2014; Erdogan and Stilman 2014), also concerned with tool creation, propagates physical constraints to ensure a symbolic planner considers only acceptable configurations of objects.[1]

As for the virtual objects, most production-system architectures (e.g., Klahr, Langley, and Neches 1987) support rules that introduce new symbols, with associated attribute values, in elements they add to working memory. However, they do not elevate their creation to the architectural level or make theoretical claims about the way such objects are defined, processed, and used by other mechanisms. Our extended framework associates virtual objects with concept instances that reside in belief memory, so that any conceptual rule in long-term memory can generate them during the inference process. This allows a tight integration with other components of the ICARUS architecture.

Otherwise, the paradigm most relevant to our use of virtual objects is scene understanding (e.g., Antanas et al. 2012), which attempts to infer models of the environment from images or videos. Classical approaches construct a hierarchy of entities, from edges to angles to surfaces to 3D object models (Binford 1982). ICARUS' virtual objects are directly analogous to these intermediate entities, and its calculation of derived attribute values maps directly on computations of angles and volumes in vision systems. However, work in this paradigm has focused on scene interpretation, not with goal-directed activity. Thus, although such systems might be able to describe and recognize tools like bridges and stairs, they cannot use them to achieve objectives.

[1] Brown and Sammut (2012) report a novel approach to learning tool usage by the analysis of training cases, but their research has different aims than our own.

Plans for Future Work

We have shown that the extended ICARUS can represent and reason about tools, it can construct such tools from available objects, and it can then use them to achieve its goals. Nevertheless, we must still address a number of challenges that our work to date has left unexamined. The most obvious limitations involve the system's dependence on handcrafted knowledge about composite tools.

ICARUS already includes mechanisms for learning hierarchical skills from successful problem solving (Langley et al. 2009), and we can use this ability to acquire structures for constructing bridges, staircases, and similar artifacts, as well as ones for using them after they have been created. The latter will be useful in more realistic environments that require sequences of actions for tool use, such as taking repeated steps up a staircase. These mechanisms acquire new skills from individual solutions obtained through search, so learning can be very rapid.

A more challenging hurdle involves the acquisition of concepts that recognize composite tools. Here we plan to draw on another extension to ICARUS (Li et al. 2012) that, when it uses a problem solution to create a new skill, also defines a new conceptual predicate that describes the conditions under which that skill will achieve the relevant goals. These conceptual rules may be disjunctive or even recursive, so the mechanism should be able to produce concepts for recognizing bridges, staircases, and other tools that may have arbitrary numbers of components.

However, we can best take advantage of this ability by separating the issues of tool construction and tool use. If we present an ICARUS agent with a problem that it can solve with an existing configuration of objects, say two blocks that cover a gap, it could learn both a hierarchical skill for using that configuration and a concept that recognizes similar 'bridge' configurations in the future. Given such knowledge, it could then solve, and learn from, new problems that require the construction of a bridge before its traversal. This decomposition is not strictly necessary, but inventing the bridge concept from scratch would require more search than determining how to build one after having used another.

These are certainly not the only challenges that remain before we have a mature account of tool construction and use. For instance, numeric simulation of durative operators, as in Langley et al.'s (2016) PUG architecture, seems relevant to determining whether an agent can use a tool to achieve its goals. The ability to interleave planning, execution, and monitoring is also important in settings where tools are not fully reliable. However, the creation and use of tools is one of the distinguishing features of human intelligence, so we should not be surprised that many open problems remain.

Concluding Remarks

In this paper, we reported extensions to the ICARUS architecture that support the creation and use of tools. These included the ability to associate numeric attributes with concepts and skills, as well as calculate their values during inference, execution, and problem solving. Another augmentation let conceptual rules refer to new, complex objects that were composed from existing ones and to derive values for their numeric attributes during the process of conceptual inference. Together, these capabilities let the extended architecture not only represent and reason about tools it creates from components available in the environment, but also use those tools to achieve its goals.

We demonstrated this new functionality in two simulated environments, one that involved creating and traversing a bridge and another that required constructing and climbing a staircase. We will not claim that other approaches, such as AI planning methods, cannot handle the same tasks, but they would not represent or recognize the fact that tools played a key role in their solutions. Humans clearly exhibit this ability, and we believe that ICARUS' approach to tool creation and use has many similarities. Nevertheless, we have taken only the first steps, and future work should include demonstrations in more realistic environments and use of learning mechanisms to acquire tool-related concepts and skills.

Acknowledgements

We dedicate this work in memory of Mike Stilman, a friend and colleague who inspired us to bridge the gap between robotics and cognitive systems. This research was supported by Grants N00014-12-1-0143 and N00014-15-1-2517 from the Office of Naval Research.

References

Anderson, J. R.; and Lebiere, C. 1998. *The atomic components of thought*. Mahwah, NJ: Erlbaum.

Antanas, L.; Frasconi, P.; Costa, F.; Tuytelaars, T.; and Raedt, L. D. 2012. A relational kernel-based framework for hierarchical image understanding. In G. Gimel'farb et al., Eds., *Structural, syntactic, and statistical pattern recognition*, 171–180. Berlin: Springer.

Binford, T. O. 1982. Survey of model-based image analysis systems. *International Journal of Robotics Research* 1:18–64.

Bonasso, R. P.; Firby, R. J.; Gat, E.; Kortenkamp, D.; Miller, D. P.; and Slack, M. G. 1997. Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental & Theoretical Artificial Intelligence* 9:237–256.

Brown, S.; and Sammut, C. 2013. A relational approach to tool-use learning in robots. In F. Riguzzi and F. Elezn, Eds., *Inductive Logic Programming*, 1–15. Berlin: Springer.

Choi, D. 2011. Reactive goal management in a cognitive architecture. *Cognitive Systems Research* 12:293–308.

Coles, A. J.; Coles, A. I.; Fox, M.; and Long, D. 2012. COLIN: Planning with continuous linear numeric change. *Journal of Artificial Intelligence Research* 44:1–96.

Erdogan, C.; and Stilman, M. 2014. Incorporating kinodynamic constraints in automated design of simple machines. *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2931–2936. Chicago: IEEE Press.

Fikes, R.; and Nilsson, N. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208.

Klahr, D.; Langley, P.; and Neches, R. Eds. 1987. *Production system models of learning and development*. Cambridge, MA: MIT Press.

Laird, J. E.; Newell, A.; and Rosenbloom, P. S. 1987. Soar: An architecture for general intelligence. *Artificial Intelligence* 33:1–64.

Langley, P.; Barley, M.; Meadows, B.; Choi, D.; and Katz, E. P. 2016. Goals, utilities, and mental simulation in continuous planning. *Proceedings of the Fourth Annual Conference on Cognitive Systems*. Evanston, IL.

Langley, P.; Choi, D.; and Rogers, S. 2009. Acquisition of hierarchical reactive skills in a unified cognitive architecture. *Cognitive Systems Research* 10:316–332.

Langley, P., Laird, J. E., and Rogers, S. 2009. Cognitive architectures: Research issues and challenges. *Cognitive Systems Research* 10:141–160.

Levihn, M.; and Stilman, M. 2014. Using environment objects as tools: Unconventional door opening. *Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2502–2508. Chicago: IEEE Press.

Li, N., Stracuzzi, D. J., and Langley, P. 2012. Improving acquisition of teleoreactive logic programs through representation extension. *Advances in Cognitive Systems* 1:109–126.

Nau, D. S.; Au, T.-C.; Ilghami, O.; Kuter, U.; Murdock, J. W.; Wu, D.; and Yaman, F. 2003. SHOP2: An HTN planning system. *Journal of Artificial Intelligence Research* 20:379–404.

To, S. T.; Langley, P.; and Choi, D. 2015. A unified framework for knowledge-lean and knowledge-rich planning. *Proceedings of the Third Annual Conference on Cognitive Systems*. Atlanta, GA.

Zech, P.; Haller, S.; Lakani, S. R.; Ridgeand, B.; Ugur, E.; and Piater, J. 2017. Computational models of affordance in robotics: A taxonomy and systematic classification. *Adaptive Behavior* 25:235–271.

Flexible Goal-Directed Agents' Behavior via DALI MASs and ASP Modules

Stefania Costantini, Giovanni De Gasperis
Dip. di Ingegneria e Scienze dell'Informazione e Matematica (DISIM),
Università di L'Aquila, Coppito I-67100, L'Aquila, Italy
email: {stefania.costantini, giovanni.degasperis}@univaq.it

Abstract

This paper describes the architecture that integrates DALI MASs (Multi-Agent Systems) and ASP (Answer Set Programming) modules for reaching goals in a flexible and timely way, where DALI is a computational-logic-based fully implemented agent-oriented logic programming language and ASP modules includes solvers that allow affordable and flexible planning capabilities. The proposed DALI MAS architecture exploits such modules for flexible goal decomposition and planning, with the possibility to select plans according to a suite of possible preferences and to re-plan upon need. We present an abstract case-study concerning DALI agents which cooperate for exploring an unknown territory under changing circumstances in an optimal or at least suboptimal fashion. The architecture can be exploited not only by DALI agents, but rather by any kind of logical agent.

Introduction

Adaptive autonomous agents are capable of adapting to partially unknown and potentially changing environments (Knudson and Tumer 2011), (Jiming 2001). This requires agents to be capable of various forms of commonsense reasoning and planning over a distributed multi agent architecture. A related work based on procedural reasoning system and belief desire intention (BDI) architecture is PROPHETA (Fichera et al. 2017), an object oriented procedural Python-based multi agent framework with a declarative language approach, used to control autonomous robots. Since (Costantini 2011), we advocated agent architectures capable of smooth integration of several modules/components representing different behaviors/forms of reasoning, possibly based upon different formalisms. Therefore, the overall agent's behavior can be seen as the result of dynamic combination of these behaviors, also in consequence of the evolution of the agent's environment.

We proposed in particular to adopt Answer Set Programming (ASP) modules, where ASP (cf., among many, (Baral 2003; Leone 2007; Truszczyński 2007) and the references therein) is a successful logic programming paradigm suitable for planning and reasoning with affordable complexity; many efficient implementations of ASP solvers are freely available like: CLASP (Gebser et al. 2007), Cmodels (Lierler 2005), DLV (Leone et al. 2006b), Smodels (Elkabani, Pontelli, and Son 2004) . The DALI agent-oriented language and framework was invented, designed and developed in our research group (De Gasperis, Costantini, and Nazzicone 2014; Costantini and Tocchio 2002; 2004; Costantini 2015a); the framework has been lately augmented with a plugin for the invocation of answer set solvers so to build specific modules. The ASP modules can be exploited in agents in a variety of ways: for instance in the case of reasoning about possibility and necessity, and a greater set of reasoning contexts. We have recently enhanced the integration by adopting ASP modules for planning purposes, allowing an agent or a MAS to choose among the various plans that can be obtained by means of suitable preferences.

In this paper, we show an architecture based on DALI and ASP modules to cope with complex goals, but that can be easily generalized to other agent-oriented frameworks; goals that can take profit from the subdivision into subgoals if one of the following (or both) conditions as met:

- the instance size of the planning problem to be solved for reaching the goal is too big for efficient and timely solution, the instance can be partitioned into sub-problems and the sub-solutions can and must be re-combined/merged together;
- the goal naturally splits into sub-goals where plans can/must be devised separately, and then recombined/merged together at a later stage.

The architecture exploits not a single DALI agent but a distributed MAS (Multi-Agent System), with suitable components for generating and executing plans; it allows to distribute goals and sub-goals while controlling the generation/exploitation of solutions, and possible (even partial) replanning in case of environmental changes.

We introduce an ideal case study to show how DALI agents can cooperate in order to explore an unknown territory, such as what can happen in the real world upon occurrence of some kind of catastrophic-like disruptive events (earthquake, fire, flooding, terrorist attack), were geolocalized information can easily become obsolete in few seconds and rescue planning is needed, no matter what is the difficulty.

We propose a solution based upon a MAS instead of a monolithic software solution because we consider important that each software component, i.e. agent, should partially retain its autonomy during asynchronous event processing, in the context of agent-oriented software engineering methodologies (Gomez-Sanz and Fuentes-Fernández 2015) . In fact, in this way each agent can be enriched with high-level reasoning/control behaviors that can coexists with the planning/executing activity. The MAS solution also permits to distribute the computational effort among cloud computing facilities and embedded computers so to increase overall robustness by means of advanced features such as self-monitoring and self-diagnostic, as shown in (Bevar et al. 2012). As discussed below the MAS can be based upon a controller agent which partitions a planning problem, established certain features (e.g., related to plan selection), assigns tasks of planning, re-planning and plan execution. ASP modules are meant to be exploited for planning purposes. Qualitative aspects of the proposed solution consist in: (1) the general MAS structure, that can be customized in order to cope with real-world problems; (2) the interaction between the MAS and the ASP module(s); (3) the adoption of user preferences for choosing among possible plans.

The paper is structured as follows. In the first two sections we recall ASP and the DALI language and framework. We then present the proposed MAS architecture, and an abstract case study. Finally we discuss the proposal and conclude.

Answer Set Programming in a Nutshell

"Answer set programming" (ASP) is a well-established logic programming paradigm adopting logic programs with default negation under the *answer set semantics*, which (Gelfond and Lifschitz 1988; 1991) is a view of logic programs as sets of inference rules (more precisely, default inference rules). In fact, one can see an answer set program as a set of constraints on the solution of a problem, where each answer set represents a solution compatible with the constraints expressed by the program. For the applications of ASP, the reader can refer for instance to (Baral 2003; Leone 2007; Truszczyński 2007). However, planning is among the more suitable an successful applications of ASP , cf (Son 2017; Romero, Schaub, and Son 2017) and the references therein, were planning in ASP is analyzed even under incomplete information.

Syntactically, a program (or, for short, just "program") Π is a collection of *rules* of the form:

$$H \leftarrow L_1, \ldots, L_m, not\ L_{m+1}, \ldots, not\ L_{m+n}$$

where H is an atom, $m \geqslant 0$ and $n \geqslant 0$, and each L_i is an atom. Symbol $\leftarrow$ is usually indicated with :- in practical systems. An atom L_i and its negated counterpart $not\ L_i$ are called *literals*. The left-hand side and the right-hand side of the clause are called *head* and *body*, respectively. A rule with empty body is called a *fact*. A rule with empty head is a *constraint*, where a constraint of the form $\leftarrow L_1, ..., L_n$. states that literals $L_1, \ldots, L_n$ cannot be simultaneously true in any answer set.

Unlike a conventional logic program, a ASP program may have several answer sets, each of which represent a consistent solution to given problem and constraints, or may have no answer set at all, which means that no solution can be found. Whenever a program has no answer sets, it is said that the program is *inconsistent* (w.r.t. *consistent*). In the case of planning, each answer set (if any exists) represents a plan.

All solvers provide a number of additional features useful for practical programming, that we will introduce only whenever needed. Solvers are periodically checked and compared over well-established benchmarks, and over challenging sample applications proposed at the yearly ASP competition (cf. (Calimeri et al. 2012), (Gebser, Maratea, and Ricca 2016) for recent reports).

The DALI language: Framework and Applications

DALI (Costantini and Tocchio 2002; 2004) is an Agent-Oriented Logic Programming language, (Costantini 2015a) for a comprehensive and updated list of references. A DALI agent is triggered by several kinds of asynchronous events: external events, internal, present and past events. A DALI MAS does not explicitly requires using a global clock mechanism, but temporal logic can be implemented inside agents.

External events are syntactically indicated by the postfix *E*. Reaction to each such event is defined by a reactive rule, where the special token :>. The agent remembers to have reacted by converting an external event into a *past event* (postfix *P*). An event perceived but not yet reacted to is called "present event" and is indicated by the postfix *N*.

In DALI, **actions** (indicated with postfix *A*) may have or not preconditions: in the former case, the actions are defined by actions rules, in the latter case they are just action atoms. An action rule is characterized by the new token :<. Similarly to events, actions are recorded as past actions.

Internal events is what makes a DALI agent agent proactive. An internal event is syntactically indicated by the postfix *I*, and its description is composed of two rules. The first one contains the conditions (knowledge, past events, procedures, etc.) that must be true so that the reaction (in the second rule) may happen. Thus, a DALI agent is able to react to its own conclusions. Internal events are automatically attempted with a default internal frequency customizable by means of directives in the agent initialization file, where the frequency will depend upon the very nature of each such event, and the degree of criticality for the agent.

The DALI communication architecture implements the DALI/FIPA protocol (Foundation for Intelligent Physical Agents 2003), which consists of the main FIPA primitives, plus few new primitives which are particular to DALI. The architecture may also include a filter on communication based on ontologies and forms of commonsense reasoning, as shown in previous works.

The DALI programming environment at current stage of development (De Gasperis, Costantini, and Nazzicone 2014) offers a multi-platform folder environment, built upon Sicstus Prolog programs, shell scripts, Python scripts to integrate external applications, a JSON/HTML5/jQuery web user interface to integrate into DALI applications, with a Python/Twisted/Flask web server capable to interact with A DALI MAS at the backend. We have recently devised a cloud DALI implementation, reported in (Costantini, De

Gasperis, and Nazzicone 2017; Costantini et al. 2017). In fact, as we have since long been convinced of the potential usefulness of the DALI logical agent-oriented programming language in the cognitive robotic domain, in the abovementioned papers we have presented the extensions to the basic pre-existing DALI implementation with a number of useful new features, and in particular allow a DALI MAS to interact with robots over messages buses like ROS, YARP, Redis event broker. As shown in (Costantini, De Gasperis, and Nazzicone 2017), the DALI framework has been extended to "DALI 2.0" by using open sources packages, protocols and web based technologies. DALI agents can thus be developed to act as high level cognitive robotic controllers, and can be automatically integrated with conventional embedded controllers. The web compatibility of the framework allows real-time monitors and graphical visualizers of the underline MAS activity to be specified, for checking the interaction between an agent and the related robotic subsystem. The cloud package ServerDALI allows a DALI MAS to be integrated into any practical environment. In (Costantini et al. 2017) paper we have illustrated the new "Koiné DALI" framework, where a Koiné DALI MAS can cooperate without problems with other MASs, programmed in other languages, and with object-oriented applications. In summary, the enhanced DALI can be used for multi-MAS applications and hybrid multi-agents and object-oriented applications, and can be easily integrated into preexistent applications.

The DALI framework has been experimented, e.g., in applications for user monitoring and training, in emergencies management (like first aid triage assignment), in security or automation contexts, like home automation or processes control, and, more generally, in every situation that is characterized by asynchronous events (either simple events and/or events that are correlated to other ones even in complex patterns). An architecture encompassing DALI agents and called, F&K (Friendly-and-Kind) system (Aielli et al. 2016) has been proposed for (though not restricted to) applications the eHealth domain. F&Ks are "knowledge-intensive" systems, providing flexible access to dynamic, heterogeneous, and distributed sources of knowledge and reasoning, within a highly dynamic computational environment consisting of computational entities, devices, sensors, and services available in the Internet and in the cloud. As a suitable general denomination for systems such as F&Ks we propose "Dynamic Proactive Expert Systems" (DyPES): in fact, such systems are aimed at supporting human experts and personnel or human users in a knowledgeable fashion, so they are reminiscent of the role of traditional expert systems. However, they are proactive in the sense that such systems have objectives (e.g., monitoring patients, managing resources, exploring territories, etc.) that they pursue autonomously, requiring human intervention only when needed. They are also dynamic, because they are able to exploit not only a predefined knowledge base: rather, they are equipped with a number of reasoning modules, and they are able to locate other such modules, and the necessary knowledge and reasoning auxiliary resources. In fact, DyPESs are characterized by "Knowledge-intensity", in the sense that in general a large amount of heterogeneous information and data must be retrieved, shared and integrated in order to reason within the system's domain. DyPESs can be Cyber-Physical Systems integrating software and physical components (Khaitan and McCalley 2015), and can be able to perform Complex Event Processing, i.e., to actively monitor event data so as to make automated decisions and take time-critical actions (DALI has been in fact empowered with CEP capabilities (Costantini 2015b)).

Agents (and in particular robotic agents) have complex goals that may need to be decomposed, either hierarchically or anyway into related subgoals; moreover, such goals may change in time depending upon the interaction with the environment. Prolog-based logical agents such as DALI agents but also agents written in other agent-oriented computational-logic-based languages (e.g., AgentSpeak (Rao and Georgeff 1991; Bordini and Hübner 2010), GOAL (Hindriks 2009; 2010), 3APL (Dastani et al. 2004; Dastani, van Birna Riemsdijk, and Meyer 2005)) can devise and execute plans. However, they are not easily able to decompose goals into subgoals, evaluate (based upon preferences) alternative plans, and re-plan if needed, possibly for some subgoals only; implementing such features within a single agent would in fact make the agent code heavy to understand and execute.

We have since long equipped DALI with a plugin for invoking ASP solvers and thus executing ASP modules. When this module is used for planning, it would be possible to choose among the generated plans based upon qualitative and quantitative user preferences; the preference strategies implemented so far are: (i) shortest plan; (ii) minimal-cost plan; (iii) plan including a minimum/maximum number of a certain kind of actions; we intend to implement plan evaluation based upon preferences on resource consumption, following the principles of (Costantini and Formisano 2010; 2009; Costantini, Formisano, and Petturiti 2010).

Below we propose a DALI MAS architecture aimed at goal decomposition, sub-goal assignment, planning and replanning concerning complex goals.

The ASP-MAS Architecture

In this section we illustrate the features of the proposed architecture. The DALI MAS is intended to fulfill the so-called *bounded rationality principle* (Gigerenzer 2004), which we translate that a plan for reaching a goal shall to be devised and executed in a timely manner before a ultimate T_{max} deadline. Consequently, there is a second deadline $T_{PlanMax} < T_{Max}$ by which a plan has to be computed and selected, so that the remaining time is sufficient to execute that plan. Parameters $T_{PlanMax}$ and T_{Max} are indeed dependent of the problem domain. At the current state of development they have to be determined by the MAS-ASP designer and stay constant always during run-time phase.

We also consider the hypothesis that for each problem P proposed to the MAS, a trivial solution plan can always be computed in time T_{Pt} by using a well tested deterministic algorithm, such that T_{Pt} is a negligible time compared to T_{Ps}, which is the minimum time needed to generate an acceptable sub-optimal plan.

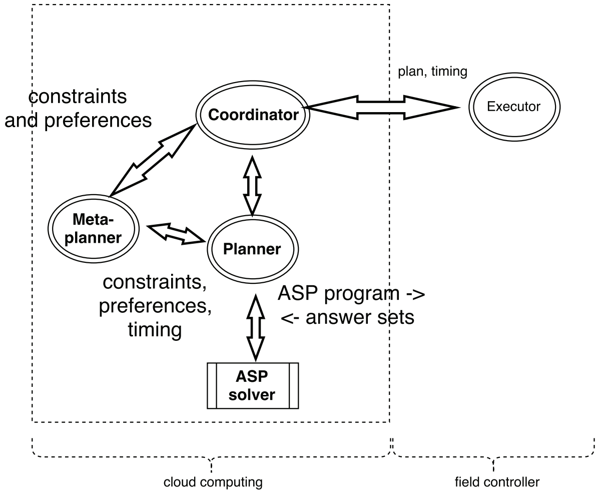


Figure 1: DALI ASP-MAS architecture: **Coordinator**, **Meta-Planner**, **Planner**, **Executor agents**. The MAS can be deployed over a cloud computing architecture, thus distributing and balancing the required computational resources. The ASP module is executed via an external solver, configurable depending on the required capabilities. The **executor** agent is supposed to actually execute the plan, possibly working "in the field", i.e., embedded in a mobile robot or some other ad-hoc facility or mechanism. Constraints can be used to codify knowledge about the environment, like obstacles, target coordinates, resources, depending on the problem domain.

Thus, given the input set $T_{PlanMax}, T_{Max}, G, N, C$, where G is the goal, N is the instance size of the problem to be solved (if applicable), C is the constraints set which models the dynamics and knowledge about the environment, the MAS operates via the following steps, not necessarily in sequence, but in parallel whenever it is possible:

(i) Decompose the overall goal into suitable subgoal;

(ii) For each subgoal, generate an a sub-plan within the $T_{PlanMax}$ deadline;

(iii) Execute the plan within the T_{Max} deadline deploying over the set of executors;
in case of failure (insufficient time to execute), maximize the length of the partially executed plan;

(iv) In case of a change of conditions in the environment, i.e. constraints change, re-plan, possibly limiting this activity to specific subgoals resulting from the partitioning.

Since each ASP module may possibly find more than one plan for given (sub-)goal, it is useful (as said before) to apply a given metrics by which a plan could be preferred to another one. The proposed DALI ASP-MAS architecture is shown in Figure 1 and the agent behaviors are here described .

- **COORDINATOR** agent: this agent synchronizes all the actions of the MAS and updates the global state of goal solving. Its task are the following.

  (a) Ensure the proper activation of the MAS and overall self checking.

  (b) Interact with the external world and whenever needed acquire new constraints for the MAS or revise the present goals.

  (c) Control the $T_{PlanMax}$ and T_{Max} deadlines.

  (d) Decompose the goal into subgoals.

  (e) For each subgoal, instantiate a **META-PLANNER** agent, possibly providing as input the preference criterion for plan selection.

  (f) receive from each **META-PLANNER** agent the sub-plan to be executed up to $T_{PlanMax}$ and deploy the overall plan to the **EXECUTOR** agents set, each is in charge of sub-plan execution within maximum time $T_{Max} - T_{PlanMax}$.

  (h) If time elapses, or new events occur, cancel the current running plan and if applicable send a replan indication to the **META-PLANNER**.

  (h) Logs all events to a log server.

- **META-PLANNER** agent, whose tasks are the following.

  (a) Receive the triggering event from the **COORDINATOR** with new constraints to start the search for a new plan.

  (b) Generate input set of constraints and specific data for the **PLANNER** agent while monitoring its performances. If **PLANNER** agent does not deliver before $T_{PlanMax} - T_{Pt}$, cancel the plan request and ask **PLANNER** to generate a trivial plan .

  (c) Apply plan selection accorded to preferences, either local or set by **COORDINATOR** agent. It also exploits the given preference criterium in order to select the plan which is closer to present preferences whenever the **PLANNER** returns more than one answer.

  (d) If requested by **COORDINATOR**, ask **PLANNER** for re-planning with updated input set of contraints.

- **PLANNER** agent, which receives as input the time constraints $T_{PlanMax}, T_{Max}, C_{\%}, N, F$ from **META-PLANNER** generate the ASP program which then generates all possible sub-plan via the ASP module, if possible within the $T_{PlanMax}$ deadline. If more than a single answer is produced by the ASP solver, it returns all available plans to the **META-PLANNER**. If no solution exists, it generates a trivial plan (if possible). The $C_{\%}$ parameter encode knowledge about the sub-optimality of the desired plan type, which coincide with the Hamiltonian plan at 100%, or refers to sub-optimal plans for lower percentages.

- **EXECUTOR**: each agent puts into action in the real world the specific sub-plan provided by the **COORDINATOR**, if possible within the T_{Max} deadline, and notifies the **COORDINATOR** upon completion. The executor agent in general executes plans (also) embodied in a physical components in a Cyber-Physical System, and/or by means of robotic elements of various kinds. In Figure 1, **EXECUTOR** is designated as "field controller" as plan execution is situated into some environment.

Summarizing, the final execution made the EXECUTORs depends on the following information:

- timing parameters, ASP program templates, static constraints imposed by the designer
- selected goals and preferences by the user
- the environment model built upon sensors perceptions which define dynamic constraints
- consistency and self-checking rules in the knowledge base
- available energy and resources, which may also have non trivial impact on hardening the constraints set.

Since in general this is a hard-NP problem, most probably only sub-optimal plans can be generated, but with a controllable desirable quality by balancing user preferences, accuracy, and weak vs. hard constraints. The resulting behavior should be similar to what a rational human expert would do in similar circumstances, with the advantage of not being limited also by human errors due to over fatigue and less concentration. So the human could dedicate himself to supervise the overall system behavior under less cognitive load stress and intervene with appropriate common sense reasoning when needed, most probably when the system is producing too many trivial plans.

Abstract Case Study

The ASP-MAS architecture presented above has been inspired and motivated by a case-study that has been actually implemented and experimented, and presented in (Costantini, De Gasperis, and Nazzicone 2015). The overall goal in the case study is to explore an unknown territory upon occurrence of some kind of catastrophic-like disruptive event (earthquake, fire, flooding, terrorist attack, etc.). The similarity comes from the idea that after such event, most of the available geo-localized information can became obsolete in a very short time and important decision have to be made in order to save lives and/or deliver rescue services. So there is a contemporary need to re-scan the territory to know were is possible to engage rescue equipments, and to generate an actually rescue plan that covers the maximum possible area were is needed. So there are places were is impossible to go (i.e. *forbidden cells*) and places were victims have to be rescued (i.e. *to_reach cells*).

For simplicity, we have modeled the territory (also called "area") as a set of a $N * N$ parts represented as chessboards, i.e., squares of cells, where some cells are marked as unreachable/forbidden, and are therefore considered as "holes" in the chessboard. This represents the fact that the agents may be notified by an external authority or by other sources of the actual impossibility of traversing that location because of some kind of obstruction/danger. The forbidden/unreachable locations, and their respective constraints set, can change in time as the scenario evolves.

For the sake of experiments, the EXECUTOR agent is embodied by a robot explorer/rescuerer[1] that each agent employs for exploration of the territory; this robot has been represented (in the case study) as a chess' knight piece, which performs knight leaps. This is to signify that a real robot (whatever its kind) will in practice have limited possibilities of movement. In this way, the problem of exploration of a single piece of territory can be modeled as a variant of the well-known "knight tour with holes" problem, for which well-known ASP solutions exist. The ultimate objective would be that of devising an Hamiltonian path, thus fully exploring the given piece of territory while skipping the forbidden squares. As however the Hamiltonian path option may results computationally intractable with reasonable instance size (already from sizes ≥ 8, or 10 using the most recent ASP more efficient solvers), we resorted to sub-optimal solutions that the MAS is capable to generate, which adopt soft constraints in order to visit each square as few times as possible.

The Knight Tour with holes problem has constituted a benchmark in recent ASP competitions, aimed at comparing ASP solvers performances. We performed a number of modifications to the original version (Calimeri and Zhou 2014) concerning: the representation of holes; the objective of devising a path which, though not Hamiltonian, guarantees a required degree of coverage with the minimum number of multiple-traversals; simple forms of loop-checking for avoiding at least trivial loops. For the sake of completeness, below is the sketch of our solution, formulated for the DLV ASP solver (Leone et al. 2006a), though it might be easily reformulated for other solvers. The key modifications to the base solution are the following.

- We modified the *reached* constraint, and transformed it into a soft constraint, so as not to be forced to finding a Hamiltonian path.

```
reached(X,Y) :- move(1,1,X,Y).
reached(X2,Y2) :-
    reached(X1,Y1), move(X1,Y1,X2,Y2).
:~ cell(X,Y),
   not forbidden(X,Y), not reached(X,Y).
```

- We added a coverage-satisfaction rule, where $coverage$ denotes the required degree of coverage and *number_forbidden* the number of holes, and V is the instance size, i.e., the chessboard edge. The maximum possible coverage is 100% of the available cells, i.e., $M = V * V$, while the minimum coverage N is computed in terms of $coverage$, considering the holes. Suitable application of the *count* DLV constraint (Leone et al. 2006a) guarantees the desired coverage.

```
coverage(95).
number_forbidden(5).
cov(N) :-
   N <= #count{X,Y : reached(X,Y)} <= M,
   size(V), coverage(Z),
   number_forbidden(F),
   M = V * V, N2 = M * Z,
   N3 = N2 /100, N = N3 - F.
```

Experimental results have demonstrated the usefulness of the proposed MAS architecture, that is actually able to effectively cope with real-world instance sizes. The architecture in this case study works as follows.

[1] not necessary a robot, also a human guided ambulance, or a combination of UAV and human guided vehicles

- The COORDINATOR agent partitions the territory that must be explored into a number of (possibly overlapping) sections (chessboards) of reasonable size (maximum 10x10 cells), each one to be assigned to a META-PLANNER instance.
- Each plan to be executed (exploration to be performed) is assigned to a separate (EXECUTOR)EXLORER agent, specifically assigned to that territory section. Each instance of the META-PLANNER agent relies upon its own associated instance of the planner agent.
- different preference policies can possibly be associated with different sections of the territory to be explored, according to directions provided by the user/environment.
- The COORDINATOR will devise re-planning for each portion of the territory for which the unreachable location have changed.

Reasonable metrics measure plans returned by the ASP module in terms of: (i) number of cells that have to be visited when using coverage, (ii) length of the path, (iii) presence of loops (when the Hamiltonian constraint is released); (iv) plan cost, in case there is a specific cost associated to each cell. Preference criteria can then be defined by selecting one metric, or by combining different metrics: for instance, a criterium may consist in preferring the shortest path, if it does not exceed a certain cost.

Concluding Remarks

We have proposed an ASP-MAS architecture for flexible goal decomposition, plan formation and execution that delivers acceptable solution to complex problems under the "*bounded rationality principle*". In real application, a MAS for each (class of) goal(s) would be designed, implemented and located into the DALI cloud. In fact, all components of the MAS will be programmed according to the goal to be reached, i.e., to the problem to be solved. Each agent that needs to solve a goal refers to the suitable MAS. As mentioned, the DALI framework allows uniform access also to agents written in other languages/formalisms. So, the proposed solution is not DALI-specific but rather can be generally adopted.

References

Aielli, F.; Ancona, D.; Caianiello, P.; Costantini, S.; De Gasperis, G.; Di Marco, A.; Ferrando, A.; and Mascardi, V. 2016. FRIENDLY & KIND with your health: Human-friendly knowledge-intensive dynamic systems for the e-health domain. In *Highlights of Practical Applications of Scalable Multi-Agent Systems. The PAAMS Collection - International Workshops of PAAMS 2016, Proceedings*, volume 616 of *Communications in Computer and Information Science*, 15–26. Springer.

Baral, C. 2003. *Knowledge representation, reasoning and declarative problem solving*. Cambridge University Press.

Bevar, V.; Muccini, H.; Costantini, S.; De Gasperis, G.; and Tocchio, A. 2012. A multi-agent system for industrial fault detection and repair. In *Advances on Practical Applications of Agents and Multi-Agent Systems.*, Advances in Intelligent and Soft Computing, 47–55. Springer, Berlin Heidelberg. Paper and demo.

Bordini, R. H., and Hübner, J. F. 2010. Semantics for the jason variant of agentspeak (plan failure and some internal actions). In Coelho, H.; Studer, R.; and Wooldridge, M., eds., *ECAI 2010 - 19th European Conference on Artificial Intelligence, Proceedings*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, 635–640. IOS Press.

Calimeri, F., and Zhou, N.-F. 2014. Knight tour with holes ASP encoding. See http://www.mat.unical.it/aspcomp2013/files/links/benchmarks/encodings/aspcore-2/22-Knight-Tour-with-holes/encoding.asp.

Calimeri, F.; Ianni, G.; Krennwallner, T.; and Ricca, F. 2012. The answer set programming competition. *AI Magazine* 33(4):114–118.

Costantini, S., and Formisano, A. 2009. Modeling preferences and conditional preferences on resource consumption and production in ASP. *Journal of of Algorithms in Cognition, Informatics and Logic* 64(1).

Costantini, S., and Formisano, A. 2010. Answer set programming with resources. *Journal of Logic and Computation* 20(2):533–571.

Costantini, S., and Tocchio, A. 2002. A logic programming language for multi-agent systems. In *Logics in Artificial Intelligence, Proceedings of the 8th Europ. Conf.,JELIA 2002*, LNAI 2424. Springer-Verlag, Berlin.

Costantini, S., and Tocchio, A. 2004. The DALI logic programming agent-oriented language. In *Logics in Artificial Intelligence, Proceedings of the 9th European Conference, Jelia 2004*, LNAI 3229. Springer-Verlag, Berlin.

Costantini, S.; De Gasperis, G.; Pitoni, V.; and Salutari, A. 2017. Dali: A multi agent system framework for the web, cognitive robotic and complex event processing. In *Proceedings of the 32nd Italian Conference on Computational Logic*, volume 1949 of *CEUR Workshop Proceedings*, 286–300. CEUR-WS.org. http://ceur-ws.org/Vol-1949/CILCpaper05.pdf.

Costantini, S.; De Gasperis, G.; and Nazzicone, G. 2015. Exploration of unknown territory via DALI agents and ASP modules. In Omatu, S.; Malluhi, Q. M.; Rodríguez-González, S.; Bocewicz, G.; Bucciarelli, E.; Giulioni, G.; and Iqba, F., eds., *Distributed Computing and Artificial Intelligence, 12th International Conference, DCAI 2015, Salamanca, Spain, June 3-5, 2015*, volume 373 of *Advances in Intelligent Systems and Computing*, 285–292. Springer.

Costantini, S.; De Gasperis, G.; and Nazzicone, G. 2017. DALI for cognitive robotics: Principles and prototype implementation. In Lierler, Y., and Taha, W., eds., *Practical Aspects of Declarative Languages - 19th International Symposium, Proceedings*, volume 10137 of *Lecture Notes in Computer Science*, 152–162. Springer.

Costantini, S.; Formisano, A.; and Petturiti, D. 2010. Extending and implementing RASP. *Fundam. Inform.* 105(1-2):1–33.

Costantini, S. 2011. Answer set modules for logical agents.

In de Moor, O.; Gottlob, G.; Furche, T.; and Sellers, A., eds., *Datalog Reloaded: First International Workshop, Datalog 2010*, volume 6702 of *LNCS*. Springer. Revised selected papers.

Costantini, S. 2015a. The DALI agent-oriented logic programming language: Summary and references 2015.

Costantini, S. 2015b. Ace: a flexible environment for complex event processing in logical agents. In Matteo Baldoni, L. B., and Dastani, M., eds., *Engineering Multi-Agent Systems, Third International Workshop, EMAS 2015, Revised Selected Papers*, volume 9318 of *Lecture Notes in Computer Science*. Springer.

Dastani, M.; van Riemsdijk, B.; Dignum, F.; and Meyer, J.-J. C. 2004. A programming language for cognitive agents goal directed 3apl. In Dastani, M.; Dix, J.; and Fallah-Seghrouchni, A. E., eds., *Programming Multi-Agent Systems, First International Workshop, PROMAS 2003, Selected Revised and Invited Papers*, volume 3067 of *Lecture Notes in Computer Science*, 111–130. Springer.

Dastani, M.; van Birna Riemsdijk, M.; and Meyer, J.-J. C. 2005. Programming multi-agent systems in 3apl. In *Multi-agent programming*. Springer. 39–67.

De Gasperis, G.; Costantini, S.; and Nazzicone, G. 2014. Dali multi agent systems framework, doi 10.5281/zenodo.11042. DALI GitHub Software Repository. DALI: http://github.com/AAAI-DISIM-UnivAQ/DALI.

Elkabani, I.; Pontelli, E.; and Son, T. C. 2004. Smodels with clp and its applications: A simple and effective approach to aggregates in asp. In *International Conference on Logic Programming*, 73–89. Springer.

Fichera, L.; Messina, F.; Pappalardo, G.; and Santoro, C. 2017. A python framework for programming autonomous robots using a declarative approach. *Science of Computer Programming* 139:36–55.

Foundation for Intelligent Physical Agents. 2003. FIPA Interaction Protocolo Specifications.

Gebser, M.; Kaufmann, B.; Neumann, A.; and Schaub, T. 2007. clasp: A conflict-driven answer set solver. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, 260–265. Springer.

Gebser, M.; Maratea, M.; and Ricca, F. 2016. What's hot in the answer set programming competition. In *AAAI*, volume 16, 4327–4329.

Gelfond, M., and Lifschitz, V. 1988. The stable model semantics for logic programming. In Kowalski, R., and Bowen, K., eds., *Proceedings of the 5th International Conference and Symposium on Logic Programming (ICLP/SLP'88)*. The MIT Press. 1070–1080.

Gelfond, M., and Lifschitz, V. 1991. Classical negation in logic programs and disjunctive databases. *New Generation Computing* 9:365–385.

Gigerenzer, G. 2004. Fast and frugal heuristics: The tools of bounded rationality. *Blackwell handbook of judgment and decision making* 62:88.

Gomez-Sanz, J. J., and Fuentes-Fernández, R. 2015. Understanding agent-oriented software engineering methodologies. *The Knowledge Engineering Review* 30(4):375–393.

Hindriks, K. V. 2009. Programming rational agents in goal. In *Multi-Agent Programming*. Springer US. 119–157.

Hindriks, K. 2010. A verification logic for goal agents. In Dastani, M. M.; Hindriks, K.; and Meyer, J.-J. C., eds., *Specification and Verification of Multi-agent Systems*. Springer.

Jiming, L. 2001. *Autonomous agents and multi-agent systems: explorations in learning, self-organization and adaptive computation*. World Scientific.

Khaitan, S. K., and McCalley, J. D. 2015. Design techniques and applications of cyberphysical systems: A survey. *IEEE Systems Journal* 9(2):350–365.

Knudson, M., and Tumer, K. 2011. Adaptive navigation for autonomous robots. *Robotics and Autonomous Systems* 59(6):410–420.

Leone, N.; Pfeifer, G.; Faber, W.; Eiter, T.; Gottlob, G.; Perri, S.; and Scarcello, F. 2006a. The dlv system for knowledge representation and reasoning. *ACM Transactions on Computational Logic* 7(3):499–562.

Leone, N.; Pfeifer, G.; Faber, W.; Eiter, T.; Gottlob, G.; Perri, S.; and Scarcello, F. 2006b. The dlv system for knowledge representation and reasoning. *ACM Transactions on Computational Logic (TOCL)* 7(3):499–562.

Leone, N. 2007. Logic programming and nonmonotonic reasoning: From theory to systems and applications. In Baral, C.; Brewka, G.; and Schlipf, J., eds., *Logic Programming and Nonmonotonic Reasoning, 9th International Conference, LPNMR 2007*.

Lierler, Y. 2005. cmodels–sat-based disjunctive answer set solver. In *International Conference on Logic Programming and Nonmonotonic Reasoning*, 447–451. Springer.

Rao, A. S., and Georgeff, M. 1991. Modeling rational agents within a BDI-architecture. In *Proceedings of the Second Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'91)*, 473–484. Morgan Kaufmann.

Romero, J.; Schaub, T.; and Son, T. C. 2017. Generalized answer set planning with incomplete information. In Bogaerts, B., and Harrison, A., eds., *Proceedings of the 10th Workshop on Answer Set Programming and Other Computing Paradigms co-located with the 14th International Conference on Logic Programming and Nonmonotonic Reasoning, ASPOCP@LPNMR 2017*, volume 1868 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Son, T. C. 2017. Answer set programming and its applications in planning and multi-agent systems. In Balduccini, M., and Janhunen, T., eds., *Logic Programming and Nonmonotonic Reasoning - 14th International Conference, LPNMR 2017, Proceedings*, volume 10377 of *Lecture Notes in Computer Science*, 23–35. Springer.

Truszczyński, M. 2007. Logic programming for knowledge representation. In Dahl, V., and Niemelä, I., eds., *Logic Programming, 23rd International Conference, ICLP 2007*, 76–88.

Perspectives on the Validation and Verification of Machine Learning Systems in the Context of Highly Automated Vehicles

Werner Damm
C. v. Ossietzky University
26111 Oldenburg, Germany

Martin Fränzle
C. v. Ossietzky University
26111 Oldenburg, Germany

Sebastian Gerwinn
OFFIS e. V.
Escherweg 2, 26121 Oldenburg

Paul Kröger
C. v. Ossietzky University
26111 Oldenburg, Germany

Abstract

Algorithms incorporating learned functionality play an increasingly important role for highly automated vehicles. Their impressive performance within environmental perception and other tasks central to automated driving comes at the price of a hitherto unsolved functional verification problem within safety analysis. We propose to combine statistical guarantee statements about the generalisation ability of learning algorithms with the functional architecture as well as constraints about the dynamics and ontology of the physical world, yielding an integrated formulation of the safety verification problem of functional architectures comprising artificial intelligence components. Its formulation as a probabilistic constraint system enables calculation of low risk manoeuvres. We illustrate the proposed scheme on a simple automotive scenario featuring unreliable environmental perception.

Modern AI and especially machine learning (ML) components are believed to be a key enabler for bringing highly automated driving functions at SAE levels 4 to 5 (SAE and others 2014) onto the market. Before such systems can be released, obtaining a rigorous guarantee of their safety is essential: systematic faults within the design (including the training phase of ML based algorithms) could have dramatic effects on the overall safety of the mass-marketed system implementations and hence also for their societal acceptance. A key challenge for this verification is the inherent uncertainty involved in object identification. To illustrate the impact of such uncertainties, consider the following artifical example of a misperception (see Fig. 1).

At time t_0, the EGO vehicle (E) has detected another vehicle v_1 on the left lane using information from a camera and RADAR sensors. At a later time instant t_1, the vehicle v_1 has closed the gap to EGO and consequently is detected still. Additionally, another vehicle v_2 has been detected at very short distance in front of EGO, while another detector has recognized the presence of a bridge in front. In this situation, EGO is confronted with the decision to either perform an overtaking manoeuvre – thereby risking a collision with v_1, or to perform an emergency brake to mitigate a potential collision with vehicle v_2. A third option would be to perform an evasive manoeuvre to the right, thereby risking a collision with a bridge pillar. Note that at t_0, the space in front of the EGO vehicle has been perceived as free. In this scenario, we assume that the time gap $t_1 - t_0$ is insufficient for a vehicle v_2 to be outside warning range at t_0 and to get to the position (and speed) perceived for v_2 at t_1, given the physical constraints on vehicle dynamics. Thus, the results of the different detectors evidently are contradictory.

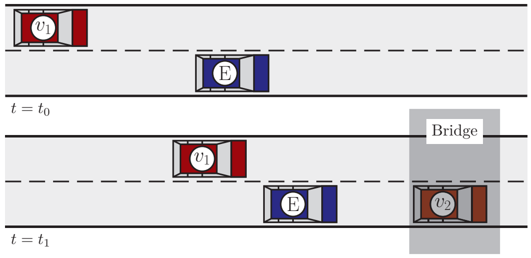


Figure 1: Example scenario. Perception of the environment is considered at two distinct time instants t_0 and t_1.

To choose an acceptable manoeuvre, a careful assessment of the risks on a vehicle level is necessary – for example by quantifying possible outcomes of a decision using injury risk scales, like AIS or ISS (MacKenzie, Shapiro, and Eastham 1985). Individual ML components, however, are traditionally evaluated using component level loss functions (Cesa-Bianchi, Conconi, and Gentile 2004). Using the common 0-1 loss ($l_{1\text{-}0}$), the resulting risk at the component level can be interpreted as bound on the probability of correctly classifying a random input (distributed according to a fixed but unknown distribution):

$$1 - \mathbb{E}[l_{1\text{-}0}] = P(\text{correctly classified}) \in [\underline{p}(\delta), \overline{p}(\delta)] \quad (1)$$

where the right hand side denotes the confidence interval as obtained from the available bounds, i.e. via cross-validation or generalisation bounds such as within the Probably-Almost-Correct (PAC) framework. These bounds in turn depend on the confidence level δ. Under the assumption that any new data (different from the training data) would be generated according to the same probability distribution which also generated the training data, a generalisation statement can be formulated and proven which provides the desired bound on the true risk.

In order to use such information to assess the risk on vehicle level, we propose a layered approach integrating the individual ML components into a constraint system which includes prior knowledge about physical properties and the functional architecture. The resulting architecture thereby combines features from probabilistic graphical models (Koller and Friedman 2009) capturing probabilistic relationships with features from non-deterministic constraint systems. We consequently employ the same definition of risk as used in reliability and utility theory (expected loss), yet permit underspecification of the probability distribution determining the expected values of interest. Among the possible instants of the underspecified distribution, we aim at calculating worst-case expectations. This permits to compute *robust* low-risk manoeuvres at runtime, whereby individual performance assessment in terms of the empirical risk at component level can be combined with the obtained constraint system to bound the overall risk at vehicle level.

In the following, we will illustrate the proposed approach on the above example, thereby illustrating its potential.

The Probabilistic Constraint System

In the example of Fig. 1, we are interested in the following analysis questions: Can we compute a robust low-risk manoeuvre for EGO at t_1, which keeps risk adequately bounded despite potentially uncertain information? Given such a robust manoeuvre, can we quantify the worst-case residual risk associated with such controller?

To answer such questions, we first construct a constraint system reflecting assumed knowledge as well as imperfect information about the underlying situation. To this end, we try to build a probabilistic system similar to a dynamic Bayesian network (Murphy and Russell 2002). In practice, we sometimes have to admit unknown dependencies not expressible in standard Bayesian networks. For such dependencies, we possess no explicit probability distribution, but can only model constraints. We illustrate such a constraint system in Fig. 2, where the functional architecture is reflected on the left side whereas information about the real world is depicted on the right side. In the following, we refer to each signal or measurement (nodes within the figure) as variables, which can be interpreted as (possibly Dirac distributed) random variables.

We assume that EGO's sensor system provides a glare detector, a bridge detector, and a vehicle detector tracking multiple vehicles. The result of each detector is an observed variable within a Bayesian network (left side of Fig. 2). As the environment and hence also the observation thereof evolves over time, each variable is also annotated with a time index t_0, t_1 (represented as shaded duplicates of the nodes). We assume the functional architecture to be given. Hence, the Bayesian Network on the left side can be constructed with known dependencies (illustrated as thin arrows). These can contain safety mechanisms like the "Fused Vehicle Detection", which employs detection of glare to improve raw object detection by situationally reducing the importance of camera-based detection. As these are only percepts of objects, corresponding real-world counterparts are modeled on the right side. Within the dynamic Bayesian network, these counterparts act as latent variables of which dependencies and probability distributions are unknown to us. Labeled test data, however, provide values for these variables on an individual data-point basis. Physical dynamical constraints, if available, furthermore restrict their possible evolution over time. Both types of information yield an overall constraint system confining possible instantiations of the unknown distributions and thus permitting to assess worst-case (across possible instantiations) residual risk of the resulting system.

Probabilistic constraints

Using access to ground truth data from manual labeling, probabilistic constraints can be derived in terms of component based performance (Eq. 1) using standard test-scores. Within our example, the performance of vehicle detection could specify a constraint on the conditional probability

$$P\left(\widehat{v}_i \mid \text{Glare } \wedge v_i \wedge \text{Bridge}\right) \in \hat{p} \pm \epsilon(\delta) \quad , \tag{2}$$

where $\hat{p}$ denotes the empirical performance, $\epsilon(\delta)$ denotes the accuracy of such an estimate depending on the confidence level δ, and v_i denotes vehicle v_i's actual presence whereas $\widehat{v}_i$ represents that v_i was detected. Analogously, fluctuations of sensor readings can be described as probability distributions conditioned on environmental states. Although some (in-)dependence connections might be known, the explicit probability distribution might be unknown. Therefore, instead of fully specifying a dynamic belief network over all discrete and continuous variables, we only collect an incomplete set of constraints of the form of Eq. (2). This necessitates an optimisation over the possible instantiations of such underspecified distributions when calculating a safe bound on the residual risk.

Dynamic constraints

In addition to such probabilistic constraints originating from individual component tests, prior knowledge about the dynamics can be incorporated (blue box 'dynamic constraints' in Fig. 2). The detected positions of vehicles v_1 and v_2 can for example be constrained via kinematic constraints of the vehicles. Such constraints can be represented as follows, where $\ell_i(t)$ denotes the position of vehicle i at time t and $\overline{v}, \overline{a}$ are intervals containing minimal and maximal values for velocity and acceleration:

$$\ell_i(t + \Delta t) \in \left(\ell_i(t) + (\Delta t\overline{v} + \frac{1}{2}\overline{a}(\Delta t)^2)\right) \tag{3}$$

Additional ontological constraints can reflect prior knowledge about the allowed relationship of detected objects.

As we have thus formalised a system involving variables on vehicle level ϕ as well as corresponding variables in the real world ψ, we can now relate systemic, real-world loss (e.g., in terms of available injury risk scales) to vehicle-level variables. As the vehicle variables include decision and actuator variables, such a loss function $l(\phi, \psi)$ evaluates the real-world severity of detecting, deciding, and acting. Note that both types of variables are collections of variables and in particular include references to different temporal instances.

Risk assessment

As mentioned earlier, we are interested in the overall risk of the designed function R as well as a situational risk R^s from

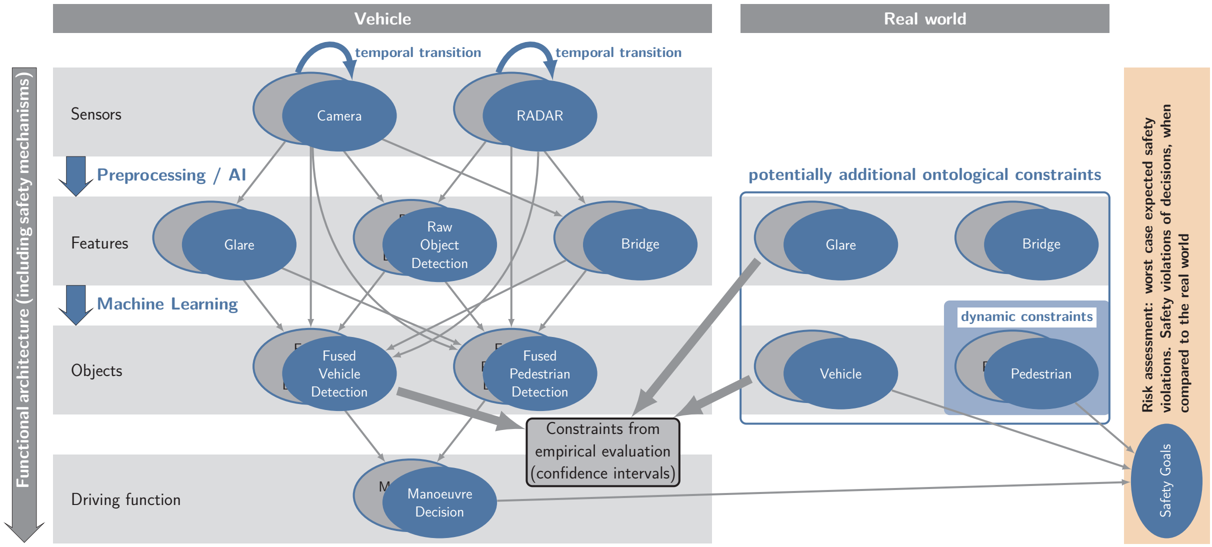


Figure 2: Structure of the probabilistic constraint system generated from the functional architecture and the constraints obtained via empirical evaluation as well as ontological and dynamic constraints. See text for more details.

which we can derive a robust low-risk manoeuvre in a given situation. Mathematically, these quantities can be described as the following expectations:

$$R = \mathbb{E}_{(\phi,\psi)}[l(\phi,\psi)],\ R^s = \mathbb{E}_{(\psi|\phi)}[l^s(\phi,\psi)] \tag{4}$$

Note that for the situational risk, we use the conditional distribution conditioned on observations obtained in the particular situation and a potentially different loss-function l^s (compared to the overall risk). More specifically, within the overall risk for the designed function, we might, e.g., want to use a binary loss function assigning $l(\phi,\psi) = 1$ if the situation was handled successfully and $l(\phi,\psi) = 0$ else. For the situational risk, we might want to use a quantitative assessment of the outcome. In contrast to the common setting of dynamic Bayesian Networks, the joint distribution $p_{\phi,\psi}$, however, is not completely given. Instead, only constraints over such a distribution are known due to equations like (2). More precisely, constraints as in (2) can be written as projections of the joint distribution using Bayes Rule:

$$\begin{aligned} &P\left(\widehat{v}_i \mid \text{Glare } \wedge v_i \wedge \text{Bridge}\right) \\ &= \frac{P\left(\widehat{v}_i \wedge \text{Glare } \wedge v_i \wedge \text{Bridge}\right)}{P\left(\text{Glare } \wedge v_i \wedge \text{Bridge}\right)}\ , \end{aligned} \tag{5}$$

where each of the constraint variables either is a variable of the vehicle domain or of the real world (see Fig. 2). As the expression above omits some of the variables defined in those domains, the corresponding expressions have to be obtained by marginalising $p_{\phi,\psi}$. The question whether the (overall or situational) residual risk meets a desired bound ϑ can be formulated as a noisy optimisation problem

$$\max_{p_{\phi,\psi}\in\mathcal{P}} \mathbb{E}_{(\phi,\psi)}[l(\phi,\psi)] \overset{?}{\leq} \vartheta,\ \max_{p_{\phi,\psi}\in\mathcal{P}} \mathbb{E}_{(\psi|\phi)}[l^s(\phi,\psi)] \overset{?}{\leq} \vartheta\ , \tag{6}$$

where the different constraints restrict the possible distributions, in the above formulation denoted by the set $\mathcal{P}$. If all variables are discrete, constraints on the distribution can directly be encoded into constraints on the distribution-values for different valuations of the vehicle or real-world variables. For continuous variables, the distribution has to be parametrised accordingly. Both types of constraints, however, can be incorporated into possibly non-linear functions g_i acting on the parametrised version of the distribution and the variables ϕ,ψ. For the empirical constraint of Eq. (2,5), such functions can be formalised as follows:

$$\begin{aligned} &C_i(P,\phi,\psi)\ \text{def.:}\ g_i(P,\phi,\psi) \leq c_i \\ &\underbrace{\frac{\int p(\phi,\psi)d((\phi\cup\psi)\setminus\{\widehat{v}_i, v_i,\ \text{Glare, Bridge}\})}{\int p(\phi,\psi)d((\phi\cup\psi)\setminus\{v_i,\ \text{Glare, Bridge}\})}}_{:=g_0(P,\phi,\psi)} \leq \underbrace{\hat{p}+\epsilon(\delta)}_{:=c_0} \end{aligned} \tag{7}$$

Using specification techniques of stochastic satisfiability modulo theory (Fränzle, Hermanns, and Teige 2008), the problem (6) can alternatively be formulated as:

$$\exists_{P:\bigwedge_i C_i(P,\phi,\psi)} \rotatebox[origin=c]{180}{R}_{\phi,\psi\sim P} : l(\phi,\psi) \overset{?}{\leq} \vartheta \tag{8}$$

Here, we collected all constraints over the distribution as well as over the variables within the conjunction $\bigwedge_i C_i$. Exploiting importance sampling for Eq. 8 (Fränzle et al. 2015), such problem can be made amenable for analysis using available tools (Fränzle, Gao, and Gerwinn 2017). To address scalability issues, one can also resort to statistical model checking (Ellen, Gerwinn, and Fränzle 2014).

Verification and situational analysis

Calculating the maximal risk as formalised in the previous section provides quantitative evidence to an overall safety verification process on vehicle level. Depending on the number of constraints with confidence statements, one can calculate an overall confidence level on the risk as well. Each confidence-based constraint holds with a certain confidence. If these can be regarded as independent, the overall confidence level is merely the product of the individual confidence levels. In case one is not willing to assume independence between the confidence-based constraints, the overall confidence level can be incorporated in a way similar to probabilistic constraints like (2). Note that such constraints also include constraints like c-approximate-independence as used in (Shalev-Shwartz, Shammah, and Shashua 2017), however we allow for even more pessimistic bounds whenever less information about the dependence is available.

The calculation of the maximal risk can also be performed in a particular situation. Instead of marginalising variables for the expected loss in (4), we can fix the valuation of vehicular variables to the observed values. The maximal risk then enables one to identify the most critical real-world situations and to choose a minimal risk manoeuvre. For our example, this facilitates inferring whether it is indeed more likely to falsely detect v_2 at time t_1 than having it not detected at time t_0. As due to the dynamic constraint, either v_2 has been missed at time t_0 and correctly classified at t_1 or the other way around, this restricts the joint distribution to assign zero probability to the other possibilities. Together with the empirical evidence constraints (e.g., marginal probabilities observing glare or the probability of bridges occurring), we can therefore calculate which of the two remaining possibilities are more likely. As such, it can be interpreted as the worst case interpretation of a Bayesian filter for dynamical systems which can be applied at each point in time. However, as worst-case configurations have to be identified, scalability of such an approach remains to be demonstrated in practice, but is outside of the scope of this short-paper.

Discussion

We presented a framework designed for computing (a) the current risk under given observations and (b) the overall risk under the given constraints and marginal probabilities arising from empirical evaluations of different machine learning components involved within the functional architecture.

Within our setting, such quantities are different from inference tasks typically considered within Dynamic Bayesian Networks. The central issue is that probability distributions need not completely be known, but can be underspecified, as illustrated by the occurrence of glare or bridges provide constraints on the marginal. In fact, earlier approaches in combining constraints with Bayesian Belief Networks were frequently restricted to representing constraints as pseudo-observations (Crowley, Boerlage, and Poole 2007) or to interpreting the standard inference scheme as constraint propagation (Pearl 1985). But both can also be combined to render the inference machinery more suited for such kind of constrained network (Gogate and Dechter 2012).

Automatically learning the structure of Bayesian Networks has also been explored (Berg, Järvisalo, and Malone 2014). In such an approach, constraints about the parameters (or structure) of the underlying graph can be considered. As it fits the network parameters such that the network best explains a given dataset, that approach does not immediately fit into our robust safety verification setting.

In our work, unknown or underspecified relations between variables of the network are understood as spanning and constraining a set of possible distributions. From a frequentist point of view compatible with quantitative safety, we would like to compute worst and best case scenarios under all possible assignments across the viable probability distributions rather than missing information about the dependency of different variables. This paper explains the pragmatics and the underlying mathematical constructions; the development of scalable tools automating such reasoning as well as their benchmarking remain issues of future work.

Acknowledgements

This research was partially funded by Deutsche Forschungsgemeinschaft within the Research Training Group 1765 “SCARE: System Correctness under Adverse Conditions”.

References

Berg, J.; Järvisalo, M.; and Malone, B. 2014. Learning optimal bounded treewidth bayesian networks via maximum satisfiability. In *Artificial Intelligence and Statistics*, 86–95.

Cesa-Bianchi, N.; Conconi, A.; and Gentile, C. 2004. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory* 50(9):2050–2057.

Crowley, M.; Boerlage, B.; and Poole, D. 2007. Adding local constraints to Bayesian networks. *Advances in AI* 344–355.

Ellen, C.; Gerwinn, S.; and Fränzle, M. 2014. Statistical model checking for stochastic hybrid systems involving nondeterminism over continuous domains. *International Journal on Software Tools for Technology Transfer*. Published online: 03 August 2014.

Fränzle, M.; Gerwinn, S.; Kröger, P.; Abate, A.; and Katoen, J.-P. 2015. Multi-objective parameter synthesis in probabilistic hybrid systems. In *International Conference on Formal Modeling and Analysis of Timed Systems*, 93–107. Springer.

Fränzle, M.; Gao, Y.; and Gerwinn, S. 2017. Constraint-solving techniques for the analysis of stochastic hybrid systems. In *Provably Correct Systems*. Springer. 9–38.

Fränzle, M.; Hermanns, H.; and Teige, T. 2008. Stochastic satisfiability modulo theory: A novel technique for the analysis of probabilistic hybrid systems. In *International Workshop on Hybrid Systems: Computation and Control*, 172–186. Springer.

Gogate, V., and Dechter, R. 2012. Approximate inference algorithms for hybrid Bayesian networks with discrete constraints. *arXiv:1207.1385*.

Koller, D., and Friedman, N. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT press.

MacKenzie, E. J.; Shapiro, S.; and Eastham, J. N. 1985. The abbreviated injury scale and injury severity score: Levels of inter- and intrarater reliability. *Medical care* 823–835.

Murphy, K. P., and Russell, S. 2002. Dynamic bayesian networks: Representation, inference and learning.

Pearl, J. 1985. A constraint propagation approach to probabilistic reasoning. In *Proceedings of the First Conference on Uncertainty in Artificial Intelligence*.

SAE, O., et al. 2014. Taxonomy and definitions for terms telated to on-road motor vehicle automated driving systems. *SAE Standard J3016* 01–16.

Shalev-Shwartz, S.; Shammah, S.; and Shashua, A. 2017. On a formal model of safe and scalable self-driving cars. *arXiv preprint arXiv:1708.06374*.

SiRoK: Situated Robot Knowledge — Understanding the Balance Between Situated Knowledge and Variability

Angel Daruna,[1] * Vivian Chu,[1] Weiyu Liu,[1] Meera Hahn,[1]
Priyanka Khante[2] Sonia Chernova,[1] Andrea Thomaz[2]
[1]Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332, USA.
[2]Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712, USA.

Abstract

General-purpose robots operating in a variety of environments, such as homes or hospitals, require a way to integrate abstract knowledge that is generalizable across domains with local, domain-specific observations. In this work, we examine different types and sources of data, with the goal of understanding how locally observed data and abstract knowledge might be fused. We introduce the Situated Robot Knowledge (SiRoK) framework that integrates probabilistic abstract knowledge and semantic memory of the local environment. In a series of robot and simulation experiments we examine the tradeoffs in the reliability and generalization of both data sources. Our robot experiments show that the variability of object properties and locations in our knowledge base is indicative of the time it takes to generalize a concept and its validity in the real world. The results of our simulations back that of our robot experiments, and give us insights into which source of knowledge to use for 31 types of object classes that exist in the real world.

Introduction

Robotics is undergoing a transition from the development of specialized, single-task robots to general-purpose platforms expected to operate in diverse and changing environments, such as hospitals and homes. Operation in unconstrained human environments introduces many new challenges, one of which is that of knowledge acquisition. On the one hand, the diversity of target environments makes it impossible to precode the robot with all the required knowledge (e.g., where the towels are kept, that a particular bowl is made of metal), requiring the robot to learn from observations on-site. On the other, information often referred to as "common sense knowledge", can be transferred across domains (e.g., towels are often found in bathrooms and closets, bowls are containers) (Speer and Havasi 2012). In this work, we examine different types and sources of such data, to understand how locally observed data and abstract knowledge can be fused to enable a robot to most effectively reason about its world.

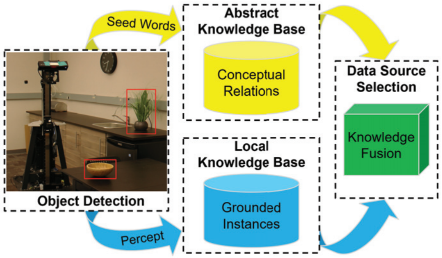


Figure 1: High-level view of SiRok framework.

As a motivating example, consider a robot placed in a new home and tasked with fetching a glass of water. One approach is for the robot to rely entirely on local observations, and to exhaustively search the environment for a glass and sink. A human visitor to the home, however, would instead be likely to first find a kitchen, then begin to open cabinets (and not drawers) in order to find the glass. This behavior would be guided by semantic, domain-independent knowledge gathered from prior experiences, and a similar capability would enable robots to more effectively adapt to new environments. However, local knowledge must also be incorporated into this reasoning, allowing adaptation to domain-specific patterns or the current state of the world, such as when the glasses have already been set out on the table, or in houses with unconventional item storage areas. In order to support a robust deployment model, we must better understand the limits of both local and abstract data.

In this work, we consider two sources of knowledge: abstract knowledge and local knowledge. We characterize *abstract knowledge* as domain-independent information that generalizes across many environments (e.g., food in typical homes can be found in the refrigerator in the kitchen). Specifically, we use commonsense information from ConceptNet (Speer and Havasi 2012) and WordNet (Miller 1995) to allow the robot to reason about novel objects and environments. We characterize *local knowledge* as information the robot has perceived in its current environment. This includes information obtained from its sensors (e.g., camera, laser, etc.), including object recognition, semantic lo-

*This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE-1650044. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

cations, and object properties. From these data sources we generate two separate knowledge bases, the Abstract Knowledge Base (AKB) and the Local Knowledge Base (LKB), which the robot uses to reason about the world. Combined, these components make up the Situated Robot Knowledge (SiRoK) framework (Fig. 1).

Our work makes the following contributions. First, we introduce a domain-independent framework for automatically retrieving common-sense knowledge for a given environment. We use object labels, obtained from object recognition, to generate seed words, which are then used to query existing semantic knowledge bases to construct a probabilistic model representing object type, location, and property data. Second, in a series of robot and simulation experiments we examine in what situations the abstract and local knowledge sources are most reliable for objects with both mutable and immutable properties. Our results show that variability is a key heuristic to take into account when evaluating knowledge sources. In particular, as variability increases, we should emphasize sources of general knowledge. For cases with extreme levels of variability, a robot should rely on direct observations or chance. Our simulations validate the trends we see in our robot experiments, and extend our conclusions to 31 different classes of objects found in real-world households.

Related Work

Numerous projects across the AI community have sought to make use of commonsense and semantic knowledge. Three large-scale commonsense knowledge networks used across a wide range of applications are WordNet (Miller 1995), ConceptNet (Speer and Havasi 2012), and ResearchCyc (Lenat 1995; Matuszek et al. 2006). WordNet consists of a collection of synsets, which connect concepts hierarchically through the *IsA* relation. WordNet also distinguishes between different senses of the same word and provides glosses, or definitions, for each sense. While WordNet is clean and hand-coded, it also lacks diversity in the types of relations it contains. ConceptNet, on the other hand, contains several dozen different relations, but it does not distinguish between word senses and is largely crowdsourced, leading to a large amount of noise. ResearchCyc uses an even larger number of relations (currently around 17,000) to connect concepts. For the purposes of this work, we choose to use data from WordNet and ConceptNet to take advantage of the complimentary benefits of each.

In other work, Zhu, et al. (Zhu, Fathi, and Fei-Fei 2014a) perform affordance prediction on a set of images by using a Markov Logic Network (MLN) (Richardson and Domingos 2006a) to represent affordance knowledge. This work also does not deal with context and used hand-selected objects and affordances in the network. In (Chen and Liu 2011), contextual noise is addressed by disambiguating the concepts in ConceptNet to enrich the WordNet senses with more diverse knowledge for improved performance on word sense disambiguation tasks. While disambiguating ConceptNet helped provide context for each of its concepts, the resulting knowledge base contained only abstract information. In contrast to this approach, (Stoica and Hearst 2004) did construct a situated knowledge hierarchy in a (nearly) automated way, however, the resulting model only included hypernyms (the *IsA* relation).

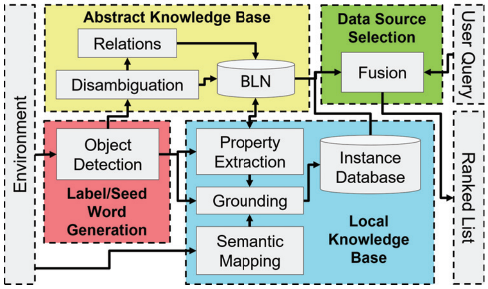


Figure 2: System architecture for the Situated Robot Knowledge (SiRoK) framework. The pipeline starts with environment data that is used to populate the AKB and LKB

Within robotics, the KnowRob (Tenorth and Beetz 2009) and RoboBrain (Saxena et al. 2014) projects are most closely related to our work. In KnowRob, the authors create a knowledge network from a variety of encyclopedic sources and represented the network using Prolog rules and the Web Ontology Language. This network is then used to repair robot task plans by filling in missing low-level details from high-level task descriptions. In RoboBrain, the authors generate a multimodal knowledge network for robotics using data collected automatically from the web. The resulting network is abstract and does not account for the domain-specific details relevant to the situational context of the robot. The RoboEarth project focused on the creation of a cloud repository of generalizable robot knowledge, including object models and robot task descriptions, that could be transferred across robot platforms and domains (Waibel et al. 2011). While these works deal with both abstract and situated knowledge, none of them investigate which knowledge source to leverage when. Our efforts focus on understanding which knowledge source a robot should use given some query (e.g. where is the plant) which may be part of a higher-level task. We conclude that the variability of a given piece of information impacts the reliability of obtaining it from either local or abstract sources.

SiRoK System Architecture

The SiRoK framework is implemented as a system of interconnected modules, which communicate using ROS. The system has three main components (Fig. 2): AKB, LKB , and Data Source Selection, each of which contains a series of subsystems that aggregate and process data. At a high-level, the pipeline begins by performing object detection, where objects in the environment are assigned an object class labels (e.g., cups, bowls, etc.). These generated class names become seed words that are used to extract information from online commonsense networks to build an AKB. These object class labels are also used during grounding, where specific object information is stored into the LKB. In Data Source Selection, the robot uses specific queries to ask

Data Types	Possible labels
Object Class	apple, banana, book, bottle, bowl, broccoli, cake, carrot, chair, clock, couch, cup, donut, fork, glass, knife, laptop, microwave, orange, oven, phone, pizza, plant, refrigerator, sandwich, sink, spoon, table, toaster, tv, vase
Colors	black, blue, brown, gray, green, orange, pink, purple, red, transparent, white, yellow
Materials	cloth, glass, metal, organic, paper, plastic, wood
Weights	light, medium, heavy
Shapes	arch, cylindrical, rectangle, spherical

Figure 3: Classes and object data in the AKB and LKB

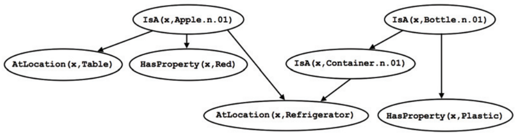


Figure 4: An example of abstract knowledge represented using a Bayesian Logic Network (BLN)

for information from AKB and LKB and fuses the results to respond to the queries. In the remainder of this section, we describe each subsystem in detail and the full system diagram can be found in Fig. 2. The colors of each component in Fig. 2 match the high-level view in Fig. 1.

Object Detection

For object detection, we used the open source real-time object detection system YOLOv2 (Redmon et al. 2016). YOLOv2 uses a convolutional neural network and computes the location and classification of each object in an image in a single pass. It does this by dividing the image into cells, calculating an objectness score and then object classification probabilities over the individual cells, it then using anchor boxes to predict the object bounding boxes. We tested YOLOv2 on PASCAL VOC2012, achieving a mAP (mean average precision) score of 73.4. For our robot experiments, we trained YOLOv2 on the subset of COCO (Lin et al. 2014) object classes which are specific to the home environment (Fig. 3). Each time the system recognizes the object, the object label, bounding box of the object, and raw rectangle segment of the object is sent to the LKB. The object labels are also passed to the AKB.

Abstract Knowledge Base

We represent the robot's AKB as a Bayesian Logic Network (BLN) (Jain, Waldherr, and Beetz 2009), a directed statistical relational model in which the variables under consideration are represented as first-order terms or predicates with arguments. BLNs allow logical constraints, represented as first-order logic rules, to be imposed on the network. Prior work in computer vision has utilized Markov Logic Networks (Richardson and Domingos 2006b), a representation that unifies Markov Random Fields and first-order logic, for modeling object attributes and affordances (Zhu, Fathi, and Fei-Fei 2014b). However, parameter learning in MLNs is an ill-posed problem (Jain, Kirchlechner, and Beetz 2007) and approximate inference is expensive even for simple queries. In contrast, BLNs are easy to train, more efficient and have scaled better to our application. Fig. 4 shows a small example BLN, which, once constructed, can be used to perform inference using likelihood weighting (Fung and Chang 2013) to answer queries such as *AtLocation*($Object_i, x$) or *HasProperty*($Object_i, x$).

To construct the BLN, we leverage information from two online sources of semantic knowledge, WordNet (Miller 1995) and ConceptNet (Speer and Havasi 2012). WordNet is a low-noise hand-crafted collection of sets of cognitive synonyms (synsets), each expressing a distinct concept (e.g., *spoon*) and related to other concepts through hypernym (the *IsA* relation, e.g., *IsA(spoon, utensil)*). ConceptNet is an auto-generated commonsense knowledge bank; it does not differentiate between word senses but groups all within a single concept node related to others through multiple possible relations. For example, for the object *mouse*, ConceptNet returns *AtLocation(mouse,office)* and *HasProperty(mouse, organic)*, highlighting the need to perform sense disambiguation to correctly parse this data.

Given seed words obtained from object recognition labels, we first perform sense disambiguation using the technique in (Tsatsaronis, Varlamis, and Vazirgiannis 2008), by finding the sense of each word that maximizes the overall similarity between the seed words (leveraging the fact that the words come from the same context). We then query WordNet and ConceptNet for semantic data related to each disambiguated word. Importantly, the seeds words not only provide a starting point for data retrieval, but together act as context for the robot's specific environment. Currently, we retrieve data for three relations, which we selected due to their usefulness in robot task execution.

- *IsA*: determines the relationship between an object and its hypernym (e.g., *IsA(bowl, container)*), allowing the robot to reason over object categories.
- *AtLocation*: determines the relationship between an object and locations in the world. (e.g., *AtLocation(bowl, sink)*, allowing the robot to query likely object locations.
- *HasProperty*: determines the relationship between an object and properties such as materials, shape, and colors (e.g., *HasProperty(bowl, ceramic), HasProperty(bowl, red)*, aiding in recognition and allowing the robot to reason about possible object uses (e.g. metal objects should not be placed in the microwave).

For each relation, we calculate a likelihood based on a weighted combination of the relation score from ConceptNet and the Explicit Semantic Analysis relatedness measure (Gabrilovich and Markovitch 2007) between the two concepts in the relation. This likelihood provides an initial estimate for the real-world probability of a given relationship and enables us to generate training evidence for BLN based on the distribution. Relations that cannot be sampled directly are inferred logically using transitive prolog rules. For additional details, see (Garrison and Chernova 2016).

Local Knowledge Base

LKB Data Structure We represent the robot's local environment through a collection of object instances, forming a

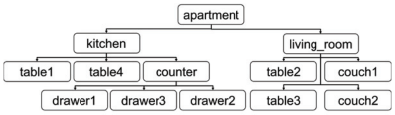


Figure 5: Topological map.

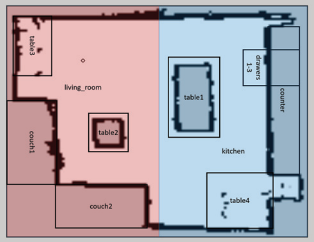


Figure 6: Metric map with an overlay of the spatial volumes associated with nodes in the topological map.

memory of encountered items, and their locations and properties. For $o \in O$, each object class out of the set of objects known to the robot (listed in Fig. 3), we store i instances of that object within the LKB, where an instance is defined as a unique object.

The LKB is implemented using PyTables and HDF5; each object class o is stored as a database, with a table generated for each object instance. For each instance, we currently store the object label, previously seen locations (pose and semantic label), image region corresponding to the bounding box from object recognition, visual information (RGB-D values), and all properties known about the instance (e.g, color, material). The resulting representation provides a scalable memory system that allows for efficient retrieval of all of its recent memories of instances.

Grounding In addition to using object recognition for object class labels (e.g., *bottle*), the robot must distinguish different instances of the same class (e.g., *red bottle* vs *yellow bottle*). The grounding component of SiRoK uses features distinct to instances of an object class to distinguish among multiple instances. This form of grounding, from here on referred to as *instance grounding*, was implemented using a K-Nearest Neighbors (KNN) classifier with a threshold distance to accommodate new instances of a class. Our implementation relies on color properties, extracted from the bounding box region of the image using the GrabCut algorithm (Rother, Kolmogorov, and Blake 2004) and uses KNN to determine whether an object is a new instance. Grounding enables the robot to perform color-based differentiation of objects, which we leverage in our study. In future work, we will expand instance grounding to incorporate spatial and temporal information about objects, as well as a wider variety of features.

Semantic Location In order to effectively generalize local information and relate it to abstract knowledge, we require a method for converting the robot's world coordinates to semantic location labels (e.g., *kitchen counter*). To provide a semantic location for an object, we utilize a hybrid map (Buschka and Saffiotti 2004), which links a topological map, consisting of a tree graph representing human domain knowledge, with a metric map of spatial locations in the environment. Fig. 5 and Fig. 6 show the topological and metric maps used in this work. The links between the topological map and metric map are expressed directly in the topological map nodes; association of each node with a volume in the metric map. This map structure enables the robot to obtain a semantic label for any 3D point that is hierarchical (e.g., object o is in a *drawer* in the *kitchen* in the *apartment*).

Property Extraction As discussed above, SiRoK enables the robot to reason about a range of object properties, including color, weight, material and shape. Through local observation, the robot is able to obtain some properties (e.g., color), while other important object characteristics (e.g., material) are very difficult to determine for existing platforms. Some complementary information, however, can often be obtained from the AKB, which obtains property information through ConceptNet. For each object, we assign a set of object properties commonly learned and used by robots (Hermans, Rehg, and Bobick 2011; Sun, Bo, and Fox 2013; Sinapov et al. 2014). These include color, shape, material, and weight. The individual values that each object can take on (e.g. blue, heavy, metal, etc.) can be found in Fig. 3.

While color is obtained using a simple color classifier, we hand-label the shape and weight of the objects. With the current state of the art we assume that these properties can be obtained easily with good accuracy via existing machine learning algorithms and the use of pre-trained classifiers (Chu, Fitzgerald, and Thomaz 2016; Sun, Bo, and Fox 2013; Sinapov et al. 2014). Future work will include exploration of the objects using the robot's arm and visual information from the RGB-D camera to learn the object properties. However, material still remains to be one of the harder properties to be learned. In this work, we can leverage a human in the environment to extract the material properties of the objects.

In its existing form, the BLN contains far too many property edges to simply verify each one with the human. Thus we present an algorithm, which takes the existing BLN generated from ConceptNet and WordNet, and actively selects a subset of property relations to verify with the human. This results in a pruned representation that is consistent with the specific objects in the current environment.

We first modify the BLN to include inter-property edges. For all properties in the BLN, we add an edge if a relation exists between them in ConceptNet. We then generate three tables. $T_{material}$: all material properties present in our BLN (i.e., holds a relation with *Material* in the ConceptNet). For the next two tables, we use the association index in ConceptNet, a measure between 0 to 1 of how related two words are. $T^{O_N}_{assoc}$: holds all the association indices between an O_N and every property belonging to that object (we ignore properties with index $<$ 0.07).$T_{interprop}$: Let P_O be a set such that each $p \in P_O$ is a property of O, this table holds the inter-property association indices between any two properties in P_O.

Next, we systematically pick the properties to query an expert for verifications. For each object, we query the expert about property, $p \in P_O$ with the highest association index in $T^{O_N}_{assoc}$. If it is verified *true* and exists in $T_{material}$, then all other material properties belonging to that object are assumed to be *false* and are not queried. We can also assume the predecessors of that property are true for O_N (e.g., if Aluminum is true, then Metal can be assumed true). For the successors, we assume their *hasProperty* relations are true (e.g., Metal true, then Opaque true), but need to query the successors with an *IsA* (e.g., if Metal true, still need to ask about Aluminum). If a node in this *isA* set is verified to be *true*, the rest are assumed to be *false*.

Next, query with the a property with the minimum inter-property association index with p, to ask the most different question next. Repeat this process until all the properties are verified as *true/false*. We construct an expert-verified BLN, *vBLN*, with all verified *true* properties. For evaluation we will look to compare this verified BLN with a ground truth BLN with a dissimilarity index, $I_{dissimilarity}$, defined as:

$$\frac{\text{Uncommon edges between ground truth and vBLN}}{\text{Total number of unique edges in ground truth and vBLN}}$$

Data Source Selection

SiRoK uses knowledge from the AKB and LKB to handle object *queries* related either to (1) what the object is, (2) where it is located, or (3) what properties it has. Within the AKB, the BLN is queried for *IsA*, *AtLocation*, and *HasProperty* information, and the results sorted by probability value. The LKB answers *AtLocation*, and *HasProperty* queries by using the stored outputs from semantic mapping and property classification, returning a ranked list of the most frequently encountered property. We note that, in general, location and property information have different characteristics. A specific object is likely to change location, possibly even frequently, whereas most of the properties we consider, such as color, are likely to change less often. Locations and properties also often generalize across instances (e.g., cups of the same color or cups stored in the same cabinet), but this depends on the variability of the object. In the next section, we evaluate how our inference performs across these different data types.

Robot Experiments

To evaluate the SiRoK system and examine the relative applicability of abstract knowledge and local knowledge, we designed a series of experiments testing the robot's ability to predict object locations and properties. Our test environment resembles a simple apartment containing furniture and different use areas, as seen in Fig. 6. For all experiments, we use the robot platform, Prentice (Fig. 1). Prentice is an omnidirectional mobile robot and has a horizontally mounted lidar for navigation and a Microsoft Kinect2 RGB-D camera mounted on a pan/tilt unit for visual sensing.[1]

Building the Knowledge Bases

We populate an AKB by using the 31 possible class labels shown in Fig, 3 to seed a BLN using ConceptNet and WordNet. As described in *SiRoK System Architecture*, these class labels come from the COCO image dataset that are associated with kitchen and living rooms. We removed one label, hot dog, due to WordNet disambiguating hot dog to sandwich. This is due to WordNet characterizing that hot dogs are sandwiches, which is partially true (i.e., a hot dog is a piece of meat between bread). Future work will address how to take into account words that are part of the same hypernym hierarchy. The constructed BLN contains 257 nodes and 358 edges.

To gather data for the LKB, we used the following experimental steps: (1) put object(s) in our testing environment, (2) allow the robot to observe the environment and update the LKB, (3) update the state of the object(s) in our environment, then repeat this process for the desired number of observations. After each observation, we evaluate the accuracy for finding objects or naming object properties on a fixed test set. To populate the semantic locations, we provide an expert labeled semantic map that correlates to the described scene in Fig. 6. We use a color classifier to label each object in the test environment and the BLN for the object material. The average classifier accuracy is 70% and average clarifications needed for object property is 2.

If a human is available, SiRoK has the option to interactively validate properties in the BLN. We performed 84 clarifications to prune 50 edges in the vBLN from 195 property edges using the human-verification algorithm mentioned in Section III-C.4. While this is a large number of clarifications, during a deployment such queries could occur over a length of time (multiple days) as the robot spends time learning about its environment. Moreover, our algorithm is currently limited by ConceptNet. ConceptNet lacks rich inter-property knowledge (i.e. if an *apple* is sweet, one can assume it is also *juicy*) and the notion of classes (i.e. *sweet, sour, spicy, tangy* all belong to the same class of *taste*), the number of queries is large. However, knowledge of *material* class and good inter-property knowledge, it fared well for *bottle* where only 3 queries were asked for 9 properties or only 1 for 5 properties of *cup*. The final dissimilarity score of the vBLN to ground truth object properties is 0.11 (6 edges difference). This means that the BLN is only 6 edges (an edge is between an object and a property) away from the ground truth and managed to learn 50 out of the total 53 edges from the ground truth.

Experiments

We break down this section into two experiments: (1) finding objects in the scene and (2) determine the properties of objects. For both, we hypothesize that the role of variability in object options is a primary factor in deciding when to use abstract vs. locally learned knowledge. If an object moves around more frequently, we should rely on reasoning about where we might find the object as opposed to remembering where was the last time or most frequently seen location. For object properties, we expect to see a similar trend.

[1] Note that we do not evaluate *IsA* queries on the robot due to the highly abstract nature of the data. *IsA* results are reported in the simulation section.

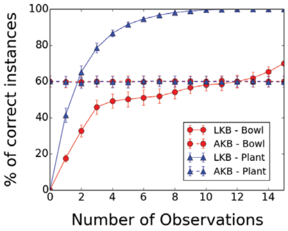


Figure 7: Average accuracy (AKB vs. LKB) across 5000 permutations to predict the top 3 locations of *potted plant* and *bowl*

Finding Objects For object location, we collect two separate sets of observations, one using a *bowl* and one using a *potted plant*. The two objects can be seen in Fig 1. For each object we collected 20 observations of the objects in various locations in the kitchen and living room. We determined the locations for each object using two different distributions for each object, one with more movement and one with less movement. The *bowl* was on the higher end of a variability spectrum (table1: 20%, table4: 20%, counter: 12%, table2: 12%, table3: 12%, drawer1: 12%, drawer2: 12%), while the *potted_plant* object was on the lower end (table3: 50%, table2: 25%, counter: 25%). Each time an object is detected by the robot, the object's semantic location is written to LKB.

To test and compare AKB and LKB, we randomly select 25% of the observations to leave out as the test set. This results in five observations in the test set and 15 in the train set. We test the accuracy AKB and LKB incrementally by introducing each observation separately. Specifically, we ask AKB and LKB to predict the location of the 5 observations in the test set after seeing one observations, two observations, and so on. AKB and LKB predict the locations by providing a ranked list of possible locations as described in *Data Source Selection*. We randomly select 5000 different permutations of the observation order and report the average accuracy and standard deviation to account for orderings effects. Note that the AKB is generated prior to seeing the observations as it represents general domain-free knowledge, so the accuracy of the AKB does not change over observations.

The results of this test can be seen in Fig. 7 where the robot turns the top three locations from its ranked list (simulating if the robot were allowed to look at three different locations to find the object). We can see that for the *potted plant*, the LKB reaches 80% accuracy by the fourth observation. However, for the *bowl*, the overall accuracy of the LKB reaches only 65% for top three locations, which is only slightly better than chance. When comparing AKB to LKB, it is clear that in cases where there is low variability in the current environment, learning about the object's location is superior to using general knowledge. However, for the bowl, where locations are more varied, the AKB does a better job of reasoning where in general might bowls be located. Furthermore, for both cases, when there is little to no knowledge of the scene, AKB still offers some insight to where the object might be located as opposed to LKB. We observed the same trends when testing the top-one results, although with lower overall performance rates.

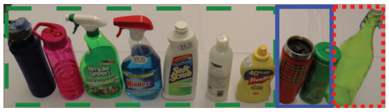


Figure 8: The bottle outlined in long green dashes, solid blue lines, and dotted red lines are plastic, metal, and glass respectively. The bottles are colored from left-to-right as blue, pink, green, blue, white, white, yellow, red, green, and green.

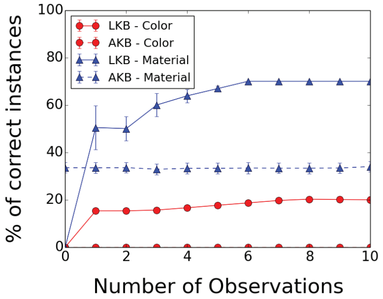


Figure 9: Average accuracy (AKB vs. LKB) across 5000 permutations for predicting the top property of 10 different bottles for two different properties (color and material)

Object Properties As described in *Data Source Selection*, object properties are fixed to a specific instance. As a result, we test the robot's ability to predict object properties by using a fixed test set that is also the observation set. As the robot observes its environment over time (similar to how one gets acquainted to a new environment), all of the objects in the environment will be added to its observation set. We select objects of the same class type (e.g., all bottles), to determine if knowledge properties of specific objects can provide insight on the general class of objects. Similar to object locations, we hypothesize that the variability of possible values for a property affects when and how we use our knowledge base. As a result, we select bottles with varying levels of variance within its properties (i.e., color is highly variable while materials is not). For this specific experiment, we selected 10 *bottles* (Fig. 8). Specifically, they ranged in color (green: 3, blue: 2, white: 2, yellow: 1, red: 1, pink:1) and materials (plastic: 7, metal: 2, glass: 1) with color more variable and material less.

The results of the test across the 10 bottles can be found in Fig. 9 for both color and material. We limit the AKB and LKB to just one guess as opposed to three for locations because for object properties, there is a higher threshold for errors. While searching three different locations in a home environment might take slightly longer, it is not unreasonable or dangerous for the robot to do so. On the other hand, predicting that an object is not metallic and putting it in the microwave could have dire consequences. As expected, the LKB performs poorly at predicting highly varied object properties. This makes intuitive sense as knowing that one

Class	Location				Color				Material				Type
	1		15		1		15		1		15		
	AKB	LKB	AKB	LKB	AKB	LKB	AKB	LKB	AKB	LKB	AKB	LKB	AKB
Apple	13	15	13	63	73	29	73	40	0	100	0	0	Food, Produce, Edible fruit, Fruit, ...
Banana	10	13	10	37	47	36	47	47	0	100	0	0	Food, Produce, Edible fruit, Fruit, ...
Book	0	8	0	23	0	14	0	20	100	100	100	100	Product
Bottle	3	6	3	20	0	20	0	27	73	27	73	73	Container, Vessel
Bowl	3	9	3	27	0	13	0	20	0	24	0	0	Stadium
Broccoli	17	16	17	60	80	69	80	80	0	100	0	0	Food, Produce, Solid
Cake	7	7	7	33	0	16	0	20	0	100	0	0	Patty, Food, Dish
Carrot	0	10	0	33	53	40	53	53	0	100	0	0	Plant organ, Plant part
Chair	17	8	17	40	0	14	0	20	33	28	33	33	Instrument
Clock	0	41	0	100	0	13	0	20	0	28	0	0	Instrument
Couch	0	19	0	70	0	13	0	20	0	32	0	0	Coloring material, Covering
Cup	13	5	13	37	0	14	0	20	20	21	20	20	Food, Drug, Agent, Fluid
Donut	7	4	7	30	0	21	0	33	0	100	0	0	Food, Doughnut, Solid
Fork	0	8	0	37	0	14	0	20	0	36	0	0	Article, Cutlery
Glass	10	4	10	30	0	16	0	20	0	27	0	0	Methamphetamine, Drug, Agent
Knife	0	8	0	13	0	15	0	20	0	40	0	0	Instrument
Laptop	0	10	0	30	0	22	0	40	33	33	33	33	Machine
Microwave	0	31	0	87	0	34	0	40	0	55	0	0	Commodity, (Home, Kitchen) appliance
Orange	0	7	0	27	0	33	0	33	0	100	0	0	Coloring material
Oven	77	32	77	100	0	40	0	53	0	61	0	0	Commodity, (Home, Kitchen)
Phone	0	9	0	23	0	16	0	27	0	34	0	0	Language unit
Pizza	3	14	3	40	0	24	0	33	0	100	0	0	Food, Dish
Plant	23	8	23	37	0	14	0	20	0	40	0	0	Building complex
Refrigerato	0	39	0	100	33	34	33	40	0	50	0	0	Commodity, Home appliance
Sandwich	30	11	30	37	13	22	13	33	0	100	0	0	Food, Dish
Sink	30	27	30	100	0	34	0	40	100	50	100	100	Container, Vessel, Cesspool, Excavation
Spoon	13	10	13	20	0	10	0	13	67	33	67	67	Container, Article, Cutlery
Table	20	13	20	40	7	16	7	27	27	22	27	27	Food, Board
Toaster	0	11	0	43	0	37	0	47	0	52	0	0	Commodity, (Home, Kitchen) appliance
Tv	0	25	0	100	0	29	0	40	0	32	0	0	Television, Medium
Vase	17	19	17	30	0	14	0	20	40	31	40	40	Container, Vessel

Note: Darker shading equates to higher scores. top three location, top one color property, top one material property

Figure 10: Accuracy of all class labels for location, color, material, and type.

cup is blue does not guarantee the next is blue. For material, the LKB performs well at predicting material as it captures that most bottles in the environment are plastic.

However, it is when we look at the color, that we gain interesting insight about object properties. We see that the AKB follows a slightly different trend than we observed in the object location experiment. We expected that with highly varying properties that the AKB could provide more insight than the LKB. However, if we look deeper at the results of the AKB, we discover that for the class bottle, the AKB has no prediction for color. We believe this points to an important distinction between the variability of an object property and the variability of an object location. When an object's property can take on almost any value (e.g., bottles can be pretty much any color), general knowledge offers little to no insight as to what property the object might have. Furthermore, this situation is also difficult for LKB to learn as the best we can hope for is chance. This suggests that for certain object properties, the only approach to predicting object properties that are highly variable is to remember the exact properties of the instance or perform directly reasoning using lower level features of the object. For both location and properties, we variability effects various accuracy levels of the AKB and LKB. To fully understand the extent in which this insight can be extended to a larger number of classes and properties, we perform a simulated experiment that looks at the variance accuracies across all described classes.

Simulated Evaluation

We exhaustively evaluate how different sources of information impact the various queries listed in *Abstract Knowledge Base* using a similar procedure and experimental setup for each query to *Building the Knowledge Bases*. Specifically, we populated a simulated world of object instances, and randomly assigned attribute values (seen in Fig. 3) and locations (seen in Fig. 6). Properties and locations were made class specific to better capture the real-world (e.g., no couch instances could be located in a drawer and televisions cannot be made of paper). While the rules set in simulation may not capture the rules of a specific real-world environment, they do capture the relationship between class variability and LKB accuracy and can be viewed as a unique layout of a specific home.

Evaluation Metric and Results

To test each query type, we start with a set of simulated instances. This set is taken as the true state of the world. Then a set of world state observations are created by randomly selecting locations and properties for each instance in the world and repeating the process for the number of world state observations. This set of world state observations were used as actual data for the LKB to process and store. To validate our hypothesis in *Experiments*, the evaluation was done similarly to that of the robot experiment where we report the top three locations and top one property. For the last query, object types (*IsA*), was tested by comparing the results of the returned values to three sets of human generated labels base on common sense for the home environment (e.g., *IsA*(Apple, Fruit) is true whereas *IsA*(Bowl, Stadium) is false).

In *Experiments*, we see a limited view of object locations and classes. By doing the simulated evaluation, we can look at if the trends seen in the robot experiment were reflected in the 31 different class types. The results of this evaluation are in Fig. 10. The table shows the accuracy of the AKB and LKB for location, color, and material by class. They are further broken down into accuracy values after seeing one observation vs seeing all 15 observations. The table also includes the different *IsA* relations for each object class.

We can see that several of the trends observed in the robot experiment hold true. For example, ovens, which are less variable in location, have a higher initial AKB accuracy than the LKB. The LKB learns the oven location perfectly after 15 observations. In general, color, which varies highly does poorly for both ABK and LKB unless the object has a notion of a color (e.g., carrot and broccoli). We see that the AKB does well on the material property if the class has a typical material it is made out of (e.g., books, sink, spoon). We test this on an aggregate scale in the next section. For the *IsA* queries, the average accuracy of the relations was 72%. Between the three sets of human labels, there was an 83.17% average pairwise percent agreement.The accuracy values between all three users were within 2% of each other. We can look at Fig. 10 to see that this accuracy can be reflected in the labels produced. It correctly identifies useful types such as apple is a food and bottle is a container. The few cases where *IsA* does not perform well can be seen with bowl being related to stadium and glass to drug.

Role of Variability

The results show that taking into account variability of local knowledge history will be essential for reasoning about new situations. The general trend is that as variability increases, a discount factor should be used to emphasize sources of general knowledge that are resistant to such effects. Fig. 11 was generated by categorizing each simulation output seen in Fig. 10 as either low (1-3 alternatives), medium (4-6 alternatives), or high (7+ alternatives) variability and averaging

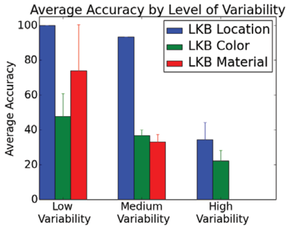


Figure 11: Relationship between LKB accuracy and variability.

all the results for each category. It shows that as variability increases, the LKB accuracy drops. For extreme levels of variability similar to in *Object Properties*, even a general knowledge systems fails. In these situations, a robot should rely on direct observations or chance.

Conclusions

In this work we introduce the SiRoK framework and systematically evaluate it through robot experiments and simulation. We use SiRoK to better understand the trade offs between general knowledge bases that store symbols and concepts and local knowledge bases that store perceptual data. We find that variability is a key heuristic to take into account when evaluating knowledge. In future works, we hope to find methods of fusing the disparate knowledge sources, improving the quality of the BLN in our AKB, and utilizing the *IsA* query.

References

Buschka, P., and Saffiotti, A. 2004. Some notes on the use of hybrid maps for mobile robots. In *Proc. of the 8th Int. Conf. on Intelligent Autonomous Systems*, 547–556.

Chen, J., and Liu, J. 2011. Combining ConceptNet and WordNet for Word Sense Disambiguation. In *International Joint Conference on Natural Language Processing*, 686–694.

Chu, V.; Fitzgerald, T.; and Thomaz, A. L. 2016. Learning object affordances by leveraging the combination of human-guidance and self-exploration. In *2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 221–228.

Fung, R., and Chang, K.-C. 2013. Weighing and Integrating Evidence for Stochastic Simulation in Bayesian Networks.

Gabrilovich, E., and Markovitch, S. 2007. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. In Veloso, M. M., ed., *International Joint Conference on Artificial Intelligence*, 1606–1611.

Garrison, H., and Chernova, S. 2016. Situated structure learning of a bayesian logic network for commonsense reasoning. *CoRR* abs/1607.00428.

Hermans, T.; Rehg, J.; and Bobick, A. 2011. Affordance prediction via learned object attributes. In *International Conference on Robotics and Automation: Workshop on Semantic Perception, Mapping, and Exploration*.

Jain, D.; Kirchlechner, B.; and Beetz, M. 2007. Extending markov logic to model probability distributions in relational domains. In *KI 2007: Advances in Artificial Intelligence*. Springer. 129–143.

Jain, D.; Waldherr, S.; and Beetz, M. 2009. Bayesian Logic Networks. Technical report, Technische Universität München, München.

Lenat, D. B. 1995. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM* 38(11):33–38.

Lin, T.; Maire, M.; Belongie, S. J.; Bourdev, L. D.; Girshick, R. B.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: common objects in context. *CoRR* abs/1405.0312.

Matuszek, C.; Cabral, J.; Witbrock, M. J.; and DeOliveira, J. 2006. An introduction to the syntax and content of cyc. In *AAAI Spring Symposium: Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering*, 44–49. Citeseer.

Miller, G. A. 1995. Wordnet: A lexical database for english. *Commun. ACM* 38(11):39–41.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788.

Richardson, M., and Domingos, P. 2006a. Markov logic networks. *Machine Learning* 62(1-2):107–136.

Richardson, M., and Domingos, P. 2006b. Markov logic networks. *Machine learning* 62(1-2):107–136.

Rother, C.; Kolmogorov, V.; and Blake, A. 2004. ”grabcut”: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* 23(3):309–314.

Saxena, A.; Jain, A.; Sener, O.; Jami, A.; Misra, D. K.; and Koppula, H. S. 2014. Robobrain: Large-scale knowledge engine for robots. *arXiv preprint arXiv:1412.0691*.

Sinapov, J.; Schenck, C.; Staley, K.; Sukhoy, V.; and Stoytchev, A. 2014. Grounding semantic categories in behavioral interactions: Experiments with 100 objects. *Robotics and Autonomous Systems* 62(5):632 – 645. Special Issue Semantic Perception, Mapping and Exploration.

Speer, R., and Havasi, C. 2012. Representing General Relational Knowledge in ConceptNet 5. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*.

Stoica, E., and Hearst, M. A. 2004. Nearly-Automated Metadata Hierarchy Creation. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 117–120.

Sun, Y.; Bo, L.; and Fox, D. 2013. Attribute based object identification. In *2013 IEEE International Conference on Robotics and Automation*, 2096–2103.

Tenorth, M., and Beetz, M. 2009. Knowrobknowledge processing for autonomous personal robots. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 4261–4266. IEEE.

Tsatsaronis, G.; Varlamis, I.; and Vazirgiannis, M. 2008. Word Sense Disambiguation with Semantic Networks. In Sojka, P.; Horák, A.; Kopeček, I.; and Pala, K., eds., *Text, Speech, and Dialogue*, 219–226. Springer.

Waibel, M.; Beetz, M.; Civera, J.; d'Andrea, R.; Elfring, J.; Galvez-Lopez, D.; Häussermann, K.; Janssen, R.; Montiel, J.; Perzylo, A.; et al. 2011. Roboearth. *IEEE Robotics & Automation Magazine* 18(2):69–82.

Zhu, Y.; Fathi, A.; and Fei-Fei, L. 2014a. Reasoning About Object Affordances in a Knowledge Base Representation. In *European Conference on Computer Vision*.

Zhu, Y.; Fathi, A.; and Fei-Fei, L. 2014b. Reasoning about object affordances in a knowledge base representation. In *European conference on computer vision*, 408–424. Springer.

Teaching Virtual Agents to Perform Complex Spatial-Temporal Activities

Tuan Do, Nikhil Krishnaswamy, James Pustejovsky
Department of Computer Science
Brandeis University
Waltham, MA 02453 USA
{tuandn, nkrishna, jamesp}@brandeis.edu

Abstract

In this paper, we introduce a framework in which computers learn to enact complex temporal-spatial actions by observing humans, and outline our ongoing experiments in this domain. Our framework processes motion capture data of human subjects performing actions, and uses qualitative spatial reasoning to learn multi-level representations for these actions. Using reinforcement learning, these observed sequences are used to guide a simulated agent to perform novel actions. To evaluate, we render the action being performed in an embodied 3D simulation environment, which allows evaluators to judge whether the system has successfully learned the novel concepts. This approach complements other planning approaches in robotics and demonstrates a method of teaching a robotic or virtual agent to understand predicate-level distinctions in novel concepts.

Motivation

The community surrounding "learning from (human) observation" (LfO) studies how computational and robotic agents can learn to perform complex tasks by observing humans (Young and Hawes 2015). Work in this area can be traced back to reinforcement learning studies by (Smart and Kaelbling 2002) or (Asada, Uchibe, and Hosoda 1999), which closely resembles the way humans learn. Children, as early as 14 months old, can imitate adults in a variety of tasks, such as *turning on and off a light-box*, and can even interpret the intentions behind actions and consider all constraints involved (Gergely, Bekkering, and Király 2002).

Most robots developed in the previous decades have shipped with pre-installed programs, limited to a set of predefined functionalities. Learning approaches in the robotics community seek to move toward smarter and more adaptable robots, for the following reasons, among others:

- Consumer desire for mobile or household assistant robots that can perform multiple tasks with a flexible apparatus, such as multiple grasping arms (Bogue 2017). Robots with behavioral robustness can learn from a wider range of experiences by interacting with humans in a dynamic environment (Hawes et al. 2017).
- Advances in deep learning have afforded robotic agents a high-level understanding of embedded semantics in multiple modalities, including language, gesture, object recognition, and navigation. This increases the circumstances and modalities available for robotic learning.

Event recognition and classification have achieved recent relevance in human communication with robotic agents (Paul et al. 2017). Meanwhile, lexical computational semantic approaches to events (e.g., Pustejovsky (1995), Pustejovsky and Moszkowicz (2011)) make it clear that event semantics are compositional with their arguments.

We have previously presented an approach toward facilitating human communication with a computational agent, using a rich model of events and their participants (Pustejovsky, Krishnaswamy, and Do 2017). Formally, we have devised a semantic framework using *Multimodal Semantic Simulations (MSS)*, which can be used to encode events as programs in a dynamic logic with an operational semantics. Computationally, we have been looking at event representation through sequential modeling, using data from 3-dimensional video captures, to distinguish between different event classes (Do and Pustejovsky 2017a). In this work, we aim to bridge the gap between these two lines of research by proposing a methodology to learn programmatic event representations from linguistic and visual event representations.

Linguistic event representation in our framework is modeled as a verbal subcategorization in a frame theory, a la Framenet (Baker, Fillmore, and Lowe 1998), with thematic role arguments. However, we also account for *extra-verbal factors* in our event type distinction. For example, we consider *A moves B toward C* and *A moves B around C* to be different event types and we learn each event type as a separate action.

Our visual event representation comprises visual features extracted from tracked objects in captured videos or virtual object positions saved from a simulation environment. Both types of feature represent information visible to humans and observable by a machine in an object state. Using these data points and sequences, machines can observe humans performing actions through processing captured and annotated videos, while humans can observe machines performing actions through watching simulated scenes.

Programmatic event representation can be based on formal event semantics or on features that can direct simulated or robotic agents to perform an action with an object of given properties. From a human perspective, the distinction between learning to recognize and learning to perform an action might be obvious. However from a machine's perspective, these two tasks might require different learning methods. Our work aims to demonstrate that given an appropriate framework, it is feasible to map between them, in a manner similar to the way humans actually learn: by matching actions to observations.

In this paper, (1) we discuss related work in AI that focuses on the learning of action and object models, including our own past studies; (2) we discuss several technologies and machine learning methodologies that provide the foundation for our experiments; (3) we discuss our ongoing experiment to learn actions; (4) we discuss our evaluation scheme and possible extensions to our framework.

Related Work

Work on action and object representation can generally be divided into two types of approaches: bottom-up approaches and top-down approaches.

Bottom-up approaches include both unsupervised and supervised feature-based learning. Work such as (Duckworth et al. 2016; Alomari et al. 2017) aims for unsupervised co-learning of object and event representations in the same step, and introduced the notion of a learned *concept* as an abstraction of feature spaces. In such a framework, "learnable" concepts are any distinctions meaningful to a human, such as a facial expression, color, object property, or action distinction, and these categories can then be assigned labels based on their commonly-occurring features. Notable supervised learning studies include (Koppula, Gupta, and Saxena 2013), which jointly models the human activities and object *affordances*, or attached behaviors which the object either facilitates by its geometry (which we term Gibsonian) (Gibson, Reed, and Jones 1982), or for which it is intended to be used (which we term "telic") (Pustejovsky 1995). Such a model could be used to distinguish longer activities by means of labeling sub-activities and object affordances: for example, labeling a "meal preparation" and its different subtasks based on understanding the objects involved at each step.

The foundation of our embodied event simulation is the modeling language known as VoxML (Visual Object Concept Modeling Language) (Pustejovsky and Krishnaswamy 2016). We encode verbal programs into a dynamic logic format from which we can conduct programmatic planning of complex events from atomic subevents. This is a top-down approach in which verbs are encoded with their subevent structures into programmatic "voxemes," or visual instantiations of lexemes which can then be visualized and enacted by an agent in a virtual environment. Subevent programs may themselves be linked to other voxemes, allowing for condition satisfaction, as in Figure 1, where "touching" is defined as the EC (externally connected) relation in RCC (Region Connection Calculus (Randell et al. 1992)). This is underspecified and may be further constrained by relative orientations between the two objects involved: x and y.

We aim to unify the two broad types of approaches outlined above using a form of *apprenticeship learning*, wherein a learning model observes an expert demonstrating the task that we want it to learn to perform. We propose a model, cf. (Abbeel and Ng 2004), in which reinforcement learning is used as a backbone for planning, while estimating a reward function as measuring the progression of the event-actions to be learned.

$$\left[\begin{array}{l} \textbf{touching} \\ \text{LEX} = \left[\begin{array}{l} \text{PRED} = \textbf{touching} \end{array}\right] \\ \text{TYPE} = \left[\begin{array}{l} \text{CLASS} = \textbf{config} \\ \text{VALUE} = \textbf{EC} \\ \text{ARGS} = \left[\begin{array}{l} \text{A}_1 = \textbf{x:3D} \\ \text{A}_2 = \textbf{y:3D} \end{array}\right] \\ \text{CONSTR} = \textbf{nil} \end{array}\right] \end{array}\right]$$

Figure 1: Sample voxeme: [[TOUCHING]]

Background

Simulators

VoxSim Our simulated environment is built in **VoxSim** (Krishnaswamy and Pustejovsky 2016), a semantically-informed visual event simulator built on top of the Unity game engine (Goldstone 2009). VoxSim contains a 3D agent capable of manipulating objects in the virtual world by creating parent-child relationships between the objects and its joints to simulate grasping. Assuming the simulated agent's skeleton is isomorphic to the joint structure of a physical robot, this then allows us to simulate events in the 3D world that represent real-world events (such as moving the virtual robot around a virtual table that has blocks on it in a configuration that is generated from the positioning of real blocks on a real table). The embodied agent can perform a set of simple actions:

- $ENGAGE$: grasp object near its end-effector.
- $MOVE(x)$: move end-effector (hand) to 3D point x, with parent limb motions calculated using inverse kinematics
- $DISENGAGE$: ungrasp current object, and retract the agent to standing position.

The simulation environment is used to demonstrate the agent's understanding of learned behavior, by enacting new behaviors over a set of virtual objects. Scenes generated by VoxSim will be used to evaluate performance of the system, as discussed later.

Simplified Simulator For the updating loops in our reinforcement learning algorithm, we want to simulate observational data similar to the real captured data faster than real-time for effective computation. As a real-time, graphics heavy simulator, VoxSim is not feasible for this portion of the task. We are aware of a few other physical simulation environments such as Gazebo[1], but as we do not focus on physical constraints in this study, so we implemented our own simplified simulator in Python.

[1] http://gazebosim.org/

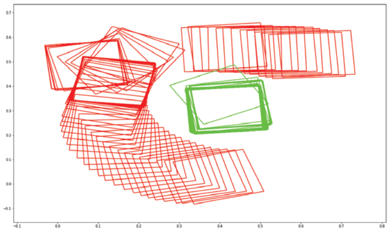


Figure 2: An event "Move A around B" projected into simulator. A is projected as a red square, B as a green square

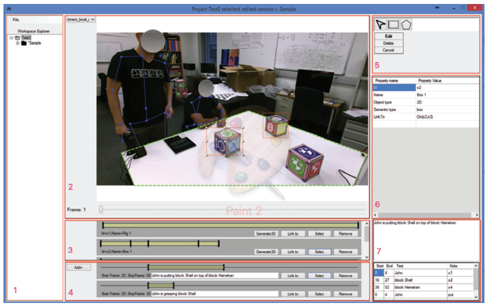
Our set of learnable actions is limited to ones that can be easily approximated in 2D space. 3D captured data is transformed into simplified simulator space by projecting it onto a 2D plane defined by the surface of the table used for performing the captured interaction. Our 2D simulator has the following features:

- Each object is represented as a polygon (or square), with a $transform$ object that stores its position, rotation, and scale.
- The space is constrained so objects do not overlap.
- Speed can be specified so that object movement can be recorded as a sequence of feature vectors interpolated from frame to frame.

Qualitative Spatial Reasoning

Qualitative spatial reasoning (QSR), a sub-field of qualitative reasoning, is considered to be formally akin to the way humans understand geometry and space, due to the cognitive advantages of conceptual neighborhood relations and its ability to draw coarse inferences under uncertainty (Freksa 1992). It is also considered a promising framework in robotic planning (Cohn and Renz 2001). QSR allows formalization of many qualitative concepts, such as *near*, *toward*, *in*, *around*, and facilitates learning distinctions between them (Do and Pustejovsky 2017b). QSR has many methods of accounting for relative vs. absolute relations, such as allowing *near* to be thresholded relative to an existing reference point (Renz and Nebel 2007), which reinforces the intuition that predicates such as *near* are inherently relative (Peters 2007). The use of qualitative predicates ensure that scenes which are semantically close have very similar feature descriptions. We use the following QSR types for feature extraction.

- CARDINAL DIRECTION measures relations between two objects as compass directions (north, northeast, etc.)
- MOVING or STATIC measures whether a point is moving or not.
- QUALITATIVE DISTANCE CALCULUS discretizes the distance between two moving points, following (Yang and Webb 2009).
- QUALITATIVE TRAJECTORY CALCULUS is a representation of motions between two objects by considering them as two moving point objects (MPOs).



Property name	Property Value
ID	o2
Name	Box 1
Object type	2D
Semantic type	box
LinkTo	On(o2,o3)

Id=o1;Name=Rig 1

Id=o2;Name=Box 1

Start Frame: 20, StopFrame: 50 John is putting block Shell on top of block Heineken

Start Frame: 20, StopFrame: 30 John is grasping block Shell

John is putting block Shell on top of block Heineken

Start	End	Text	Note
0	4	John	o1
16	27	block Shell	o2
38	52	block Heineken	o4
0	4	John	put

Figure 3: ECAT GUI showing performer interacting with recognized and annotated objects.

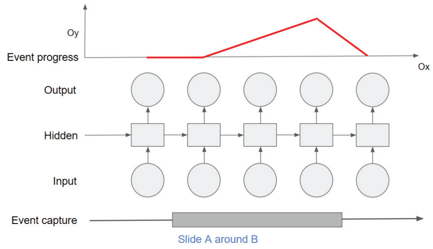


Figure 4: LSTM network producing event progress function

Event Annotation Framework

We use an event capture and annotation tool developed in our lab, ECAT (Do, Krishnaswamy, and Pustejovsky 2016), which employs Microsoft Kinect® to capture performers interacting with objects in a blocks world environment. Objects are tracked using markers fixed to their sides. They are then projected into three dimensional space using Depth of Field (DoF). Performers are also tracked using the Kinect® API, which provides three dimensional inputs of their joint points (e.g., wrist, palm, shoulder).

Learning Framework

Sequential Learning In this study, we consider a version of Long-short term memory (LSTM) (Hochreiter and Schmidhuber 1997) that processes sequential inputs to a sequence of output signals. LSTM has found utility in a range of problems involving sequential learning, such as speech and gesture recognition. Inputs are the feature vectors taken from action captures or from the simplified simulator and output is a function that corresponds to the progress of an event. In particular, we create a function that takes a sequence S of feature vectors, current frame i and action e: $f(S, i, e) = 0 \leq q_i \leq 1$

The training set of sequential captured data is passed through an LSTM network, which is fitted to predict a linear progressing function. At the start or outside of an event span, the network produces 0, whereas at the end, it produces 1.

Reinforcement Learning The objective of the embodied agent is to generate a sequence of actions to attain a maximum reward, whereas our reward corresponds to how closely the produced object movement resembles movement of objects in the training data. Visual (tracked) information is used to evaluate performance of the system.

Currently, the action space is continuous. Therefore, planning is carried out by selecting the action at step k (u_k) based on the current state of the system ($X_k \in R^n$). A stochastic planning step is parameterized by policy parameters $\theta : u_k \sim \pi_\theta(u_k|x_k)$.

This type of parameterized reinforcement learning policies is best solved by using policy gradients (Gullapalli 1990; Peters and Schaal 2008). Here, we use the REINFORCE algorithm (Williams 1992), for its effectiveness in policy gradient learning.

We consider two versions of REINFORCE, which carry out planning in continuous and discrete search spaces, respectively. For continuous space, we propose using a Gaussian distribution policy $\pi_\theta(u|x) = Gaussian(\mu, \sigma)$. For simplicity, the dimensions of μ and σ are the same as the degrees of freedom in our simplified simulator (2 dimensions for position and 1 dimension for rotation). An artificial neural network (ANN) will be used to produce values μ and σ. The set of weights in our ANN is the parameter θ from the REINFORCE algorithm, learned with gradient descent.

For discrete space, we again use a qualitative reasoning method. Specifically, the searching space for the *transform* of the target location could be separated into two spaces, for (X, Y) coordinates and rotation r. The searching space for (X, Y) could be discretized according to cardinal direction and quantized distance.

A searching method employing simple random search with back-up is used as baseline to evaluate performance of the progress learner. We will present some preliminary results from this searching method.

Experiments

Here we describe our experimental setup and evaluation plans.

Experiment

We aim to use the learning framework outlined above for teaching an agent to perform a set of actions where it interacts directly with a single object while the other objects stay relatively static and the interaction takes place over a continuous span.

1. An agent moves {object A} **closer to** {object B}
2. An agent moves {object A} **away from** {object B}
3. An agent moves {object A} **past** {object B}
4. An agent moves {object A} **next to** {object B}
5. An agent moves {object A} **around** {object B}

This set of actions differ only in their prepositional adjuncts, which describe different motion trajectories. Thus for this experiment, the learning problem is reduced to one of motion paths.

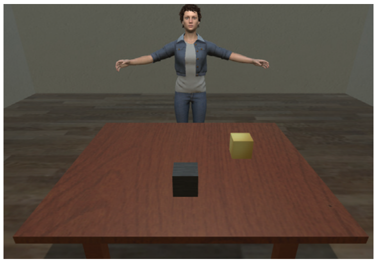


Figure 5: Visualizer implemented in Unity

These actions are, however, generally classified into different event types. Using the treatment from (Pustejovsky 1991), an action such as "moves {object A} **next to** {object B}" is an *achievement*, which means it has a logical culmination or duration. Other actions do not have a defined ending, though for "moves {object A} **closer to** {object B}," this action is ended at the point when "{object A} is **next to** {object B}." From a cognitive point of view, recognition of these action types, except possibly for **move next to**, requires consideration of the trajectory as well as the start and ending points of the objects involved. For example, **closer to** conceptually involves change of distance between the start and the ending position of the moving object relative to the static object, but a complex motion path could lead to misinterpretation of the action. **Closer to**, therefore, strongly indicates a trajectory of the moving object toward the static object.

By grouping the learning of different event types together, we aim to examine the capability of a single learning framework that to learn multiple event types. The reason is rather obvious: we, as humans, can learn all of these actions without prior knowledge of different action types.

For each action type, we are capturing 40 sessions of two different performers. Block positions are randomized at the start. We mark the beginning and end of the captured action and give it a textual description.

We generate frame-by-frame feature vectors by employing the set of aforementioned QSR features: cardinal direction and qualitative distance between objects' positions and frame-to-frame difference; qualitative trajectory for each object and frame-to-frame difference. These features are used only for the sequential model to predict event progress, whereas we use objects' parameters (positions and rotations) across consecutive frames as state of the system X_k.

Evaluation

Human evaluation will be carried out on action demonstrations generated by both the 2D simulator and our lab's 3D visualizer, VoxSim (Figure 5). In VoxSim, we create a testbed scene with blocks on a table, similar to the setup used in video captures. For each randomized configuration of objects (block positions and rotations), we command the virtual agent to perform one of the actions, and the scene is recorded for evaluators to judge its performance.

Our human-driven evaluation method aims to help answer the following questions:

1. Does the virtual agent learn the concept in question? Re-

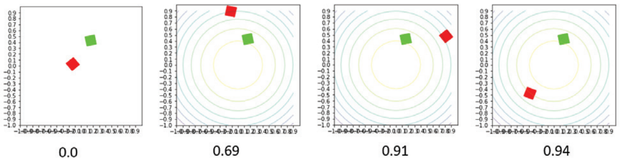


Figure 6: A correct demonstration of “Move red block around green block.”

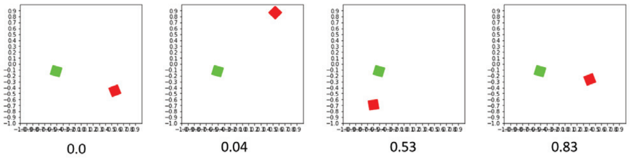


Figure 7: A wrong demonstration of “Move red block around green block.” The value beneath each frame is value predicted by the progress learner.

flected by average score given to a demonstration when annotators know the action label.

2. Can the virtual agent make distinctions between learned actions? Reflected by confusion matrix when annotators have to label the action performed in a scene.
3. Will evaluation scores on the 2D simulator significantly differ from those on the 3D visualizer?
4. Can we use the feedback from human evaluation to improve the learned model? Generated demonstrations with feedback scores complement real, captured data, and in some sense are better than learning by demonstration, in that they provide a rigorous way to include negative samples.

Evaluations of this type using VoxSim-generated scenes have already been conducted in (Krishnaswamy 2017; Krishnaswamy and Pustejovsky 2017), using Amazon Mechanical Turk to crowdsource judgments. Human judgments of a scene are given as “acceptable” or “unacceptable” relative to the event’s linguistic description.

Preliminary results

Preliminary runs of the system with brute-force searching show that the progress learner can help to generate correct demonstrations (Fig. 6), but sometimes produces deviations (Fig. 7), probably because of the lack of negative training samples. We hope that incorporating feedback from evaluators will improve the overall performance of the learner.

We also provide a quantitative breakdown of a small-scale human evaluation in Table 1. Two annotators (college students) are asked to give scores from 0 to 10 and are also asked to give comments on any video they graded between 3 and 7 (higher scores are considered better). **Evaluator Disparity** is the average of the absolute values of the differences between scores given by two annotators over the demonstrations of a particular action.

Action Type	Average Score	Evaluator Disparity
Slide Closer	5.4	1.57
Slide Away	6.48	2.37
Slide Next To	5.55	1.7
Slide Past	6.38	1.9
Slide Around	2.75	1.03

Table 1: Evaluation

Evaluator comments provide some insight into bad demonstrations. Typical comments on *Slide Next To* include “Need to be even closer”, while on *Slide Closer To* a typical comment is “The blocks touched.” That suggests some confusion between these two actions, which requires a method to help distinguish them. Three reasons are given by evaluators for low scores on *Slide Around* demonstrations: the movement being not smooth, one or more additional steps needed for completion, and many cases where the algorithm does not generate the proper trajectory.

Code, experimental and evaluation results can be found on GitHub[2]. Complete experimental results will be forthcoming at that address.

Conclusion

Two different lines of research may be extended from this framework. One involves a learning mechanism for more complex actions, such as “make a row from given objects,” and one involves learning the “manner of motion” of actions.

Learning complex actions from simpler actions requires an additional semantic framework for objects and actions. For example, to learn “make a row from given objects” given observations of 2-unit and 3-unit rows, the learner needs to be equipped with the concept of *recursion*, the concept of a composite object made from elementary objects (e.g. the size and shape of the composite object), and other abstract concepts, such as object axis and extension of a structure along said axis.

Learning the manner aspect of actions requires a finer-grained treatment of object affordances. For example, for the learner to distinguish “rolling a bottle” and “sliding a bottle,” we need to equip it with a reasoning mechanism to determine how an object’s pose and position dictate its affordances. VoxML, the underlying platform to the VoxSim system, supports modeling these types of affordance distinctions, so reference to the VoxML semantics of objects and events can provide the reasoner with the mechanism for distinguishing these behavior types, as illustrated by (Krishnaswamy and Pustejovsky 2016).

Acknowledgements

We would like to thank the reviewers for their helpful comments. This work is supported by a contract with the US Defense Advanced Research Projects Agency (DARPA), Contract W911NF-15-C-0238. Approved for Public Release, Distribution Unlimited. The views expressed are those of

[2] https://github.com/tuandnvn/learn-to-perform/

the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government. We would like to thank Alex Luu and Dr. Pengyu Hong for their discussion and input on this topic. All errors and mistakes are, of course, the responsibilities of the authors.

References

Abbeel, P., and Ng, A. Y. 2004. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, 1. ACM.

Alomari, M.; Duckworth, P.; Bore, N.; Hawasly, M.; Hogg, D. C.; and Cohn, A. G. 2017. Grounding of human environments and activities for autonomous robots. In *IJCAI-17 Proceedings*.

Asada, M.; Uchibe, E.; and Hosoda, K. 1999. Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development. *Artificial Intelligence* 110(2):275–292.

Baker, C. F.; Fillmore, C. J.; and Lowe, J. B. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*, 86–90. Association for Computational Linguistics.

Bogue, R. 2017. Domestic robots: Has their time finally come? *Industrial Robot: An International Journal* 44(2):129–136.

Cohn, A. G., and Renz, J. 2001. Qualitative spatial representation and reasoning. 46:1–2.

Do, T., and Pustejovsky, J. 2017a. Fine-grained event learning of human-object interaction with lstm-crf. *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*.

Do, T., and Pustejovsky, J. 2017b. Learning event representation: As sparse as possible, but not sparser. *arXiv preprint arXiv:1710.00448*.

Do, T.; Krishnaswamy, N.; and Pustejovsky, J. 2016. Ecat: Event capture annotation tool. *Proceedings of ISA-12: International Workshop on Semantic Annotation*.

Duckworth, P.; Alomari, M.; Gatsoulis, Y.; Hogg, D. C.; and Cohn, A. G. 2016. Unsupervised activity recognition using latent semantic analysis on a mobile robot. In *IOS Press Proceedings*, number 285, 1062–1070.

Freksa, C. 1992. *Using orientation information for qualitative spatial reasoning*. Springer.

Gergely, G.; Bekkering, H.; and Király, I. 2002. Developmental psychology: Rational imitation in preverbal infants. *Nature* 415(6873):755.

Gibson, J. J.; Reed, E. S.; and Jones, R. 1982. *Reasons for realism: Selected essays of James J. Gibson*. Lawrence Erlbaum Associates.

Goldstone, W. 2009. *Unity Game Development Essentials*. Packt Publishing Ltd.

Gullapalli, V. 1990. A stochastic reinforcement learning algorithm for learning real-valued functions. *Neural networks* 3(6):671–692.

Hawes, N.; Burbridge, C.; Jovan, F.; Kunze, L.; Lacerda, B.; Mudrová, L.; Young, J.; Wyatt, J.; Hebesberger, D.; Kortner, T.; et al. 2017. The strands project: Long-term autonomy in everyday environments. *IEEE Robotics & Automation Magazine* 24(3):146–156.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Koppula, H. S.; Gupta, R.; and Saxena, A. 2013. Learning human activities and object affordances from rgb-d videos. *The International Journal of Robotics Research* 32(8):951–970.

Krishnaswamy, N., and Pustejovsky, J. 2016. Multimodal semantic simulations of linguistically underspecified motion events. In *Spatial Cognition X: International Conference on Spatial Cognition*. Springer.

Krishnaswamy, N., and Pustejovsky, J. 2017. Do you see what I see? effects of pov on spatial relation specifications. In *Proc. 30th International Workshop on Qualitative Reasoning*.

Krishnaswamy, N. 2017. *Monte-Carlo Simulation Generation Through Operationalization of Spatial Primitives*. Ph.D. Dissertation, Brandeis University.

Paul, R.; Arkin, J.; Roy, N.; and Howard, T. 2017. Grounding abstract spatial concepts for language interaction with robots. In *IJCAI-17 Proceedings*.

Peters, J., and Schaal, S. 2008. Reinforcement learning of motor skills with policy gradients. *Neural networks* 21(4):682–697.

Peters, J. F. 2007. Near sets. Special theory about nearness of objects. *Fundamenta Informaticae* 75(1-4):407–433.

Pustejovsky, J., and Krishnaswamy, N. 2016. VoxML: A visualization modeling language. In Chair), N. C. C.; Choukri, K.; Declerck, T.; Goggi, S.; Grobelnik, M.; Maegaard, B.; Mariani, J.; Mazo, H.; Moreno, A.; Odijk, J.; and Piperidis, S., eds., *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Paris, France: European Language Resources Association (ELRA).

Pustejovsky, J., and Moszkowicz, J. 2011. The qualitative spatial dynamics of motion. *The Journal of Spatial Cognition and Computation*.

Pustejovsky, J.; Krishnaswamy, N.; and Do, T. 2017. Object embodiment in a multimodal simulation. *AAAI Spring Symposium: Interactive Multisensory Object Perception for Embodied Agents*.

Pustejovsky, J. 1991. The syntax of event structure. *Cognition* 41(1):47–81.

Pustejovsky, J. 1995. *The Generative Lexicon*. Cambridge, MA: MIT Press.

Randell, D.; Cui, Z.; Cohn, A.; Nebel, B.; Rich, C.; and Swartout, W. 1992. A spatial logic based on regions and connection. In *KR'92. Principles of Knowledge Representa-*

tion and Reasoning: Proceedings of the Third International Conference, 165–176. San Mateo: Morgan Kaufmann.

Renz, J., and Nebel, B. 2007. Qualitative spatial reasoning using constraint calculi. In *Handbook of spatial logics*. Springer. 161–215.

Smart, W. D., and Kaelbling, L. P. 2002. Effective reinforcement learning for mobile robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 4, 3404–3410. IEEE.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8(3-4):229–256.

Yang, Y., and Webb, G. I. 2009. Discretization for naive-bayes learning: managing discretization bias and variance. *Machine learning* 74(1):39–74.

Young, J., and Hawes, N. 2015. Learning by observation using qualitative spatial relations. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 745–751. International Foundation for Autonomous Agents and Multiagent Systems.

Planning Hierarchies and Their Connections to Language

Nakul Gopalan
ngopalan@cs.brown.edu
Brown University

Abstract

Robots working with humans in real environments need to plan in a large state–action space given a natural language command. Such a problem poses multiple challenges with respect to the size of the state–action space to plan over, the different modalities that natural language can provide to specify the goal condition, and the difficulty of learning a model of such an environment to plan over. In this thesis we would look at using hierarchical methods to learn and plan in these large state–action spaces. Further, we would look the using natural language to guide the construction and learning of hierarchies and reward functions.

Introduction

In this work we consider the problem of robots working with humans in real world environments, and try to postulate some solutions that are feasible to solve such problems efficiently. There are many challenges that robot interacting with humans, we specify a few that we try to address in this work. The first challenge is to plan under uncertainty in large state–action spaces, which are continuous. The problem is also exacerbated as the number of manipulable objects in the environment increase, as there is a combinatorial explosion in the state–action space with each object the agent can manipulate. In this thesis we will explore hierarchical methods to solve such tasks.

The second challenge is to follow a natural language command to its goal specification. Natural language allows multiple modalities to present commands. Commands can be specified at different orders of granularity, coarse or fine, allowing a range to specify commands like “get to the library” to “take a left turn”. Further, commands can be specified with ends or means of the task as the goal. For example, an instruction to “go to the red room” is very different from “go to the red room through the long corridor.” In this thesis we will look at methods that ground natural language commands to reward functions hierarchies or plan directly, depending on the modality demanded by the natural language command.

The third challenge involves learning to solve such tasks efficiently. This involves learning hierarchies and spatio–temporal abstractions that construct the hierarchies. We are interested in looking at connections between attribute learning and option learning to construct these hierarchies. Attribute learning previously has been done using trajectories or natural language. We want to combine these ideas to learn hierarchies, which are efficient to plan over.

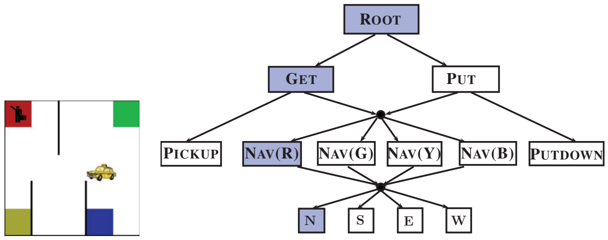
There are other challenges in robotics like partial observability, dialog, vision for robotics, task generalization, etc. which are not the focus of this thesis. In the next sections we would set up the first three challenges in detail along with our proposed solutions.

The Planning problem

When carrying out tasks in unstructured, multifaceted environments such as factory floors or kitchens, the resulting planning problems are extremely challenging due to the large state and action spaces (Bollini et al. 2012; Knepper et al. 2013). Typical planning methods require the agent to explore the state–action space at its lowest level, resulting in a search for long sequences of actions in a combinatorially large state space. For example, cleaning a room requires arranging objects in their respective places. A naive approach for arranging object might have to search over all possible states by placing all objects in all possible locations, resulting in an intractable inference problem with increasing objects.

One promising approach is to decompose planning problems in such domains into a network of independent subgoals. This approach is appealing because the decision-making problem for each subgoal is typically much simpler than the original problem. There are two ways in which the decision problem can be simplified. First, instead of selecting between actions, the agent can select between subgoals that are recursively solved, decreasing the search depth. Second, the state representation of the world can be compressed to include only information that is relevant to the current decision problem. Importantly, planning algorithms for each subproblem can be custom-tailored, allowing each goal to be solved as efficiently as possible.

We proposed *Abstract Markov Decision Process* (AMDP) hierarchies as a method for reasoning about a network of subgoals (Gopalan et al. 2017 in press), we describe the formalism briefly here. AMDPs offer a model-based hierarchical representation that encapsulates knowledge about abstract tasks at each level of the hierarchy, enabling



(a) Taxi (b) Taxi AMDP hierarchy

Figure 1: (a) The Taxi problem, where the taxi needs to drop the passenger to their goal; (b) the Taxi AMDP hierarchy, nodes indicate subgoals which are solved using an AMDP or a primitive action. The edges are actions belonging to the parent AMDP. Shaded nodes indicate which subgoals are expanded by AMDPs in a given state. In contrast, bottom-up approaches like MAXQ (Dietterich 2000) expand all nodes in the figure. These savings result in significant total planning computation gains: AMDP planning requires only 3% of the backups that MAXQ requires for the Taxi problem.

much faster, more flexible top–down planning than previous bottom–up methods like MAXQ (Dietterich 2000) or Options (Sutton, Precup, and Singh 1999). An AMDP is an MDP whose states are abstract representations of the states of an underlying *environment* (the ground MDP). The actions of the AMDP are either primitive actions from the environment MDP or subgoals to be solved. An AMDP hierarchy is an acyclic graph in which each node is a primitive action or an AMDP that solves a subgoal defined by its parent. The main advantage of such a hierarchy is that *only* subgoals that help achieve the main task need to be planned for; crucially, plans for irrelevant subgoals are never computed. Another desirable property of AMDPs is that agents can plan in stochastic environments, since each subgoal's decision problem is represented by an MDP. Moreover, each subgoal can be independently solved by any off-the-shelf MDP planner best suited for solving that subgoal.

For example, consider the Taxi problem (Dietterich 2000) shown in Figure 1a and its AMDP hierarchy in Figure 1b. The objective of the task is to deliver the passenger to their goal location out of four locations on the map. The subgoal of Get Passenger, which picks up the passenger from a source location, is represented by an MDP, with lower-level navigation subgoals, Nav(R), and a passenger-pickup subgoal, Pickup. The state space to solve the Get Passenger subgoal need not include certain aspects of the environment such as the Cartesian coordinates of the taxi and passenger. To solve this small MDP when picking up a passenger at the Red location, it is unnecessary to solve for the subpolicy to navigate to the Blue location. Our hierarchy enables a decision about which subgoal to solve without needing to solve the entire environment MDP.

In this top-down methodology, planning is performed by first computing a policy for the root AMDP for the current projected environment state, and then recursively computing the policy for the subgoals the root policy selects. In contrast a bottom up planner like MAXQ or options based planning would compute value functions over the hierarchy by processing the state–action space at the lowest level and backing up values to the abstract subtask nodes. This *bottom-up* process requires full expansion of the state–action space, resulting in large amounts of computation.

Moreover, since the tasks are abstractly defined (for example, "take passenger to blue location"), changing the task description from the "blue" to the "red" location is straightforward, and users do not have to directly manipulate the reward functions at each level of the hierarchy. This abstraction is useful in robotics, as human users can simply change the top-level task description and the required behavior will be achieved by the hierarchy.

Formally, we define an AMDP as a six-tuple ($\tilde{\mathcal{S}}$, $\tilde{\mathcal{A}}$, $\tilde{\mathcal{T}}$, $\tilde{\mathcal{R}}$, $\tilde{\mathcal{E}}, F$). These are the usual MDP components, with the addition of $F : \mathcal{S} \rightarrow \tilde{\mathcal{S}}$, a *state projection* function that maps states from the environment MDP into the AMDP state space $\tilde{\mathcal{S}}$. Additionally, the actions ($\tilde{\mathcal{A}}$) of the AMDP are either primitive actions of the environment MDP, or are associated with subgoals to solve in the environment MDP. The transition function of the AMDP ($\tilde{\mathcal{T}}$) must capture the expected changes in the AMDP state space upon completion of these subgoals. With these action and state semantics, an AMDP, in effect, defines a decision problem over subgoals for the environment MDP.

Naturally, each subgoal for a task must be solved. However, even a single subgoal might be challenging to solve in the environment MDP. Therefore, we introduce the concept of an AMDP hierarchy $H = (V, E)$, which is a directed acyclic graph (DAG) with labeled edges. The vertices of the hierarchy V consist of a set of AMDPs $\mathcal{M}$ and the set of the primitive actions $\mathcal{A}$ of the environment MDP. The edges in the hierarchy link multiple AMDPs together, with the edge label associating the action of an AMDP with either a primitive environment action or a subgoal that is formulated as an AMDP itself. Consequently, an AMDP hierarchy recursively breaks down a problem into a series of small subgoals.

We now describe planning with a hierarchy H of AMDPs. The critical property of our planning approach is to make decisions online in a top-down fashion by exploiting the transition and reward function defined for each AMDP. In this top-down methodology, planning is performed by first computing a policy for the root AMDP for the current projected environment state, and then recursively computing the policy for the subgoals the root policy selects. Consequently, the agent never has to determine how to solve subgoals that are not useful subgoals to satisfy, resulting in significant performance gains compared to bottom-up solution methods. This top-down approach does require that the transition model and reward function for each AMDP are available.

If each AMDP's transition dynamics accurately models the subgoal outcomes, then an optimal solution for each AMDP produces a recursively optimal solution for the whole problem; if the transition dynamics are not accurate, then the error associated with the overall solution can still be bounded as shown in our previous work (Gopalan et al. 2017 in press). Further, as each sub-goal has a local model, we can ground any sub-goal in the DAG depending on the

Algorithm 1 Online Hierarchical AMDP Planning

```
function SOLVE($H$)
    GROUND($H$, ROOT($H$))
function GROUND($H, i$)
    if $i$ is primitive then                          ▷ recursive base case
        EXECUTE($i$)
    else
        $s_i \leftarrow F_i(s)$                       ▷ project the environment state $s$
        $\pi \leftarrow$ PLAN($s_i, i$)
        while $s_i \notin \mathcal{E}_i$ do           ▷ execute until local termination
            $a \leftarrow \pi(s_i)$
            $j \leftarrow$ LINK($H, i, a$)            ▷ $a$ links to node $j$
            GROUND($H, j$)
            $s_i \leftarrow F_i(s)$
```

task specification as shown in the next section.

Pseudocode for online hierarchical AMDP planning is shown in Algorithm 1. Planning begins by calling the recursive *ground* function from the root of H. If node i passed to the ground function is a primitive action in the environment MDP, then it is executed in the environment. Otherwise, the node is an AMDP that requires solving. Before solving it, the current environment state s is first projected into AMDP i's state space with AMDP i's projection function F_i. Next, any off-the-shelf MDP planning algorithm associated with AMDP i is used to compute a policy. The policy is then followed until a terminal state of the AMDP is reached. Following actions selected by the policy for AMDP i involves finding the node the actions links to in hierarchy H, and then calling the ground function on that node. Note that after the ground function returns, at least one primitive action in the environment should have been executed. Therefore, after ground is called, the current state for the AMDP is updated by projecting the current state of the environment with F_i.

Planning with AMDPs shows significant improvements in planning times when compared with traditional bottom-up planners or flat planners when tested across different domains as shows in the results of (Gopalan et al. 2017 in press). We also showed a real time planning application for task and motion planning in robotics. In this demo a Turtlebot moved a block to from one room to the goal room in presence of environmental disturbances as shown in our video[1]. This is a hard planning problem with a continuous state–action space, and stochasticity in the environment. The agent shows reactive control to retrieve the block in the video as soon as it is snatched, to move the block to the goal room. For more details please refer (Gopalan et al. 2017 in press).

Hence AMDPs show significant improvements in planning times across multiple domains, even with continuous state–action spaces. Now that we have a tool to plan in large domains, we look next at natural language as an input and the different modalities of inputs, some of which would find the use of AMDP hierarchies useful.

Goal specification with Natural Language

Natural language provides an easy interface for an untrained public to work with robots. Such robots that understand natural language commands must at the very least understand goal based commands that ask the robot to achieve a certain goal configuration. Abstraction is important for achieving such goal conditions because it is much harder to map natural language to a sequence of robot control signals. Instead existing approaches map natural language commands to a formal representation at some fixed level of abstraction (Chen and Mooney 2011; Matuszek et al. 2012b; Tellex et al. 2011). While effective at directing robots to complete predefined tasks, mapping to fixed sequences of robot actions is unreliable when faced with a changing or stochastic environment. Accordingly, (MacGlashan et al. 2015) decouple the problem and use a statistical language model to map between language and robot goals, expressed as reward functions in a Markov Decision Process (MDP). Then, an arbitrary planner solves the MDP, resolving any environment-specific challenges. As a result, the learned language model can transfer to other robots with different action sets so long as there is consistency in the task representation (*i.e.*, reward functions). However natural language problem specification has different different kinds of requirements: granularity, means and ends of task solving, and temporal specification of goals.

First is the aspect of granularity. For example, a brief transcript from an expert human forklift operator instructing a human trainee has very abstract commands such as "Grab a pallet," mid-level commands such as "Make sure your forks are centered," and very fine-grained commands such as "Tilt back a little bit" all within thirty seconds of dialog. Humans use these varied granularities to specify and reason about a large variety of tasks with a wide range of difficulties. Furthermore, these abstractions in language map to subgoals that are useful when interpreting and executing a task. Moreover, MDPs for complex, real-world environments face an inherent tradeoff between including low-level task representations and increasing the time needed to plan in the presence of both low- and high-level reward functions (Gopalan et al. 2017 in press).

To address this problems, we developed an approach for mapping natural language commands of varying complexities to reward functions at different levels of abstraction within a hierarchical planning framework. This approach enables the system to quickly and accurately interpret both abstract and fine-grained commands. Our system uses a deep neural network language model that learns how to map natural language commands to the appropriate level of an AMDP planning hierarchy. By coupling abstraction level inference with the overall grounding problem, we fully exploit the subsequent AMDP hierarchy to efficiently execute the grounded tasks. To our knowledge, we are the first to contribute a system for grounding language at multiple levels of abstraction, as well as the first to contribute a deep learning system for improved robotic language understanding. The results show faster average planning times at all levels of the hierarchy when compared to a base level planner. A demo

[1] https://youtu.be/Bp3VEO66WSg

of the system can be seen here[2]. The system can accept low level commands like "go north" and high level commands like "take the block to the red room."

Next we would briefly describe other natural language grounding problems that interest us. First is problem of the means and ends of task solving, where a user might specify how to solve a task. For example the trajectory for "go to red room through the blue room" is very different from the trajectory for "go to the red room." This problem can be solved by a language model that recognizes when the means of solving a task are more important and would then plan for the task with different sets of planners. Second is the problem of temporal specification of rewards, where a command might be "go to the red room and then go to the blue room." Here, we can parse the language with Linear Temporal Logic (LTL) and create a non-Markovian reward function, where the reward functions switch as a subtask is complete. This formulation would be important to solve temporally extended tasks with multiple subgoal specifications given by the human user. Abstraction would be important in these LTL specification as solving these behavioral problems as the lowest level of abstraction might be computationally intractable. Next we look at how we might learn these abstractions.

Learning AMDP Hierarchies

The hierarchies that we looked at until now were hand designed, however an agent has to be capable of creating these hierarchical abstractions on its own in the real world. We postulate that natural language provides some clues about the levels of abstraction that a human agent might care about when working with such robots. We have two goals in this section; firstly we need to learn the local models for AMDP hierarchies; secondly a more important goal is to learn an AMDP hierarchy with language and trajectories.

To solve the first part we can use R-max (Brafman and Tennenholtz 2002) on every local model of an AMDP hierarchy. This approach will learn the level 1 models by collecting samples from the environment, but models at every higher level can be learned exactly by sampling from the models learned at level 1. This method would be sample efficient and would enlighten the trade-offs of having a precise, expensive to learn hierarchical model versus a cheap erroneous hierarchical model.

The second and more important goal is to learn an AMDP hierarchy. Konidaris 2016 uses options or temporally extended actions to learn symbols from initiation and termination sets, to create state abstractions and a higher level in the hierarchy. We believe that an important method to learn symbols can be via natural language. Matuszek et al. 2012a learned attributes of objects present in a state to model language and perception together. Borrowing ideas of attribute learning from existing literature, we can create methods to learn symbols and associated abstract states directly from demonstrations, and plan for them using AMDP hierarchies. A simpler idea to test attribute learning might be to learn object parameterized options, akin to parametrized skills, where we learn object attributes with natural language.

This learning method would satisfy most of the goals with respect to an agent in the real world that learns from natural language and example trajectories; plans in real time given a natural language command at varying degrees of granularity and temporal specification.

Conclusion

In this work we look at the problems of understanding natural language groundings, learning efficient hierarchies and planning efficiently to have a robot perform tasks real time in stochastic and large state–action spaces. Our initial results show that the planning problem can be made easier with AMDP hierarchies. We have made some inroads in the natural language grounding problems, where we can specify problems at different levels of granularity to an agent. However, we still have to make large amounts of progress in the problem of learning of a hierarchy. We believe our methods would lead to faster learning of hierarchies and shorter planning times when compared to traditional methods.

Acknowledgements

This material is based upon work supported by the National Science Foundation under grant number IIS-1637614 and the National Aeronautics and Space Administration under grant number NNX16AR61G.

References

Bollini, M.; Tellex, S.; Thompson, T.; Roy, N.; and Rus, D. 2012. Interpreting and executing recipes with a cooking robot. In *International Symposium on Experimental Robotics*.

Brafman, R. I., and Tennenholtz, M. 2002. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research* 3(Oct):213–231.

Chen, D. L., and Mooney, R. J. 2011. Learning to interpret natural language navigation instructions from observations. In *AAAI Conference on Artificial Intelligence*.

Dietterich, T. 2000. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research* 13:227–303.

Gopalan, N.; desJardins, M.; Littman, M. L.; MacGlashan, J.; Squire, S.; Tellex, S.; Winder, J.; and Wong, L. L. 2017 in press. Planning with abstract markov decision processes. In *International Conference on Automated Planning and Scheduling*.

Knepper, R.; Tellex, S.; Li, A.; Roy, N.; and Rus, D. 2013. Single assembly robot in search of human partner: Versatile grounded language generation. In *ACM/IEEE International Conference on Human-Robot Interaction Workshop on Collaborative Manipulation*.

Konidaris, G. 2016. Constructing abstraction hierarchies using a skill-symbol loop. In *International Joint Conference on Artificial Intelligence*.

[2] https://youtu.be/9bU2oE5RtvU

MacGlashan, J.; Babeş-Vroman, M.; desJardins, M.; Littman, M. L.; Muresan, S.; Squire, S.; Tellex, S.; Arumugam, D.; and Yang, L. 2015. Grounding English commands to reward functions. In *Robotics: Science and Systems*.

Matuszek, C.; FitzGerald, N.; Zettlemoyer, L.; Bo, L.; and Fox, D. 2012a. A joint model of language and perception for grounded attribute learning. *arXiv preprint arXiv:1206.6423*.

Matuszek, C.; Herbst, E.; Zettlemoyer, L.; and Fox, D. 2012b. Learning to parse natural language commands to a robot control system. In *International Symposium on Experimental Robotics*.

Sutton, R.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112(1):181–211.

Tellex, S.; Kollar, T.; Dickerson, S.; Walter, M. R.; Banerjee, A. G.; Teller, S.; and Roy, N. 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In *AAAI Conference on Artificial Intelligence*.

Learning Generalized Reactive Policies Using Deep Neural Networks

Edward Groshev,[†] **Aviv Tamar,**[†] **Maxwell Goldstein,**[‡]
Siddharth Srivatava,[*§] **Pieter Abbeel**[†]
[†] Department of Computer Science, University of California, Berkeley CA 94720
[‡] Department of Computer Science, Princeton University, Princeton, NJ 08544
[§] School of Computing, Informatics and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281

Abstract

We present a new approach to learning for planning, where knowledge acquired while solving a given set of planning problems is used to plan faster in related, but new problem instances. We show that a deep neural network can be used to learn and represent a *generalized reactive policy* (GRP) that maps a problem instance and a state to an action, and that the learned GRPs efficiently solve large classes of challenging problem instances. In contrast to prior efforts in this direction, our approach significantly reduces the dependence of learning on handcrafted domain knowledge or feature selection. Instead, the GRP is trained from scratch using a set of successful execution traces. We show that our approach can also be used to automatically learn a heuristic function that can be used in directed search algorithms. We evaluate our approach using an extensive suite of experiments on two challenging planning problem domains and show that our approach facilitates learning complex decision making policies and powerful heuristic functions with minimal human input. Video results available at goo.gl/Hpy4e3.

Introduction

In order to help with day to day chores such as organizing a cabinet or arranging a dinner table, robots need to be able plan: to reason about the best course of action that could lead to a given objective. Unfortunately, planning is well known to be a challenging computational problem: plan-existence for deterministic, fully observable environments is PSPACE-complete when expressed using rudimentary propositional representations (Bylander 1994). Such results have inspired multiple approaches for reusing knowledge acquired while planning across multiple problem instances (in the form of triangle tables (Fikes, Hart, and Nilsson 1972), learning control knowledge for planning (Yoon, Fern, and Givan 2008), and constructing generalized plans that solve multiple problem instances (Srivastava, Immerman, and Zilberstein 2011; Hu and De Giacomo 2011) with the goal of faster plan computation on a new problem instance.

In this work, we present an approach that unifies the principles of imitation learning (IL) and generalized planning for learning a *generalized reactive policy* (GRP) that predicts the action to be taken, given an observation of the planning problem instance and the current state. The GRP is represented as a deep neural network (DNN). We use an off-the-shelf planner to plan on a set of training problems, and train the DNN to learn a GRP that imitates and generalizes the behavior generated by the planner. We then evaluate the learned GRP on a set of unseen test problems from the same domain. We show that the learned GRP successfully generalizes to unseen problem instances including those with larger state spaces than were available in the training set. This allows our approach to be used in end-to-end systems that learn representations as well as executable behavior purely from observations of successful executions in similar problems.

We also show that our approach can generate representation-independent heuristic functions for a given domain, to be used in arbitrary directed search algorithms such as A* (Hart, Nilsson, and Raphael 1968). Our approach can be used in this fashion when stronger guarantees of completeness and classical notions of “explainability” are desired. Furthermore, in a process that we call “leapfrogging", such heuristic functions can be used in tandem with directed search algorithms to generate training data for much larger problem instances, which in turn can be used for training more general GRPs. This process can be repeated, leading to GRPs that solve larger and more difficult problem instances with iteration.

While recent work on DNNs has illustrated their utility as function representations in situations where the input data can be expressed in an image-based representation, we show that DNNs can also be effective for learning and representing GRPs in a broader class of problems where the input is expressed using a graph data structure. For the purpose of this paper, we restrict our attention to deterministic, fully observable planning problems. We evaluate our approach on two planning domains that feature different forms of input representations. The first domain is Sokoban (see Figure 1). This domain represents problems where the execution of a plan can be accurately expressed as a sequence of images. This category captures a number of problems of interest in household robotics including setting the dinner table. This problem has been described as the most challenging problem

[*] Part of the work was done while this author was at United Technologies Research Center

in the literature on learning for planning (Fern, Khardon, and Tadepalli 2011).

Our second test domain is the traveling salesperson problem (TSP), which represents a category of problems where execution is *not* efficiently describable through a sequence of images. This problem is challenging for classical planners as valid solutions need to satisfy a plan-wide property (namely a Hamiltonian cycle, which does not revisit any nodes). Our experiments with the TSP show that using graph convolutions (Dai et al. 2017) DNNs can be used effectively as function representations for GRPs in problems where the grounded planning domain is expressed as a graph data structure.

Our experiments reveal that several architectural components are required to learn GRPs in the form of DNNs: (1) A *deep* network. (2) Structuring the network to receive as input pairs of current state and goal observations. This allows us to 'bootstrap' the data, by training with *all pairs* of states in a demonstration trajectory. (3) Predicting plan length as an auxiliary training signal can improve IL performance. In addition, the plan length can be effectively exploited as a heuristic by standard planners.

We believe that these observations are general, and will hold for many domains. For the particular case of Sokoban, using these insights, we were able to demonstrate a 97% success rate in one object domains, and an 87% success rate in two object domains. In Figure 1 we show an example test domain, and a non-trivial solution produced by our learned DNN.

Related Work

The interface of planning and learning (Fern, Khardon, and Tadepalli 2011) has been investigated extensively in the past. The works of Khadron (Khardon 1999), Martin and Geffner (Martín and Geffner 2004), and Yoon et al. (Yoon, Fern, and Givan 2002) learn policies represented as decision lists on the logical problem representation, which needs to be hand specified. On the other hand, the literature on generalized planning (Srivastava, Immerman, and Zilberstein 2011; Hu and De Giacomo 2011) has focused on computing iterative generalized plans that solve broad classes of problem instances, with strong formal guarantees of correctness. While all of these strive to reuse knowledge made available during planning, the selection of a good *representation* for expressing the data as well as the learned functions or generalized plans is handcrafted. Feature sets and domain descriptions in these approaches are specified by experts using formal languages such as PDDL (Fox and Long 2003). Similarly, approaches such as case-based planning (Spalzzi 2001), approaches for extracting macro actions (Fikes, Hart, and Nilsson 1972; Scala, Torasso, and others 2015) and for explanation based plan generalization (Shavlik 1989; Kambhampati and Kedar 1994) rely on curated vocabularies and domain knowledge for representing the appropriate concepts necessary for efficient generalization of observations and the instantiation of learned knowledge. Our approach requires as input only a set of successful plans and their executions—our neural network architecture is able to learn a reactive policy that predicts the best action to execute based on the current state of the environment without any additional representational expressions. The current state is expressed either as an image (Sokoban) or as an instance of the graph data structure (TSP).

Neural networks have previously been used for learning heuristic functions (Ernandes and Gori 2004). Recently, deep convolutional neural networks (DNNs) have been used to automatically extract expressive features from data, leading to state-of-the-art learning results in image classification (Krizhevsky, Sutskever, and Hinton 2012), natural language processing (Sutskever, Vinyals, and Le 2014), and control (Mnih et al. 2015), among other domains. The phenomenal success of DNNs for across various disciplines motivates us to investigate whether DNNs can learn useful representations in the learning for planning setting as well. Indeed, one of the contributions of our work is a general convolutional DNN architecture that is suitable for learning to plan.

Imitation learning has been previously used with DNNs to learn policies for tasks that involve short horizon reasoning such as path following and obstacle avoidance (Pomerleau 1989; Ross, Gordon, and Bagnell 2011; Tamar et al. 2016; Pfeiffer et al. 2016), focused robot skills (Mülling et al. 2013; Nair et al. 2017), and recently block stacking (Duan et al. 2017). From a planning perspective, the Sokoban domain considered here is considerably more challenging than block stacking or navigation between obstacles. In (Tamar et al. 2016), a value iteration planning computation was embedded within the network structure, and demonstrated successful learning on 2D gridworld navigation. Due to the curse of dimensionality, it is not clear how to extend that work to planning domains with much larger state spaces, such as the Sokoban domain considered here. In that work the state space was a 2D grid world with local connectivity, making value iteration tractable. However, for Sokoban, the state must include the position of both the agent and the objects, making it much larger than a 2D grid world. While one can construct such a state space, running value iteration on it would be too slow. Another alternative is to try to embed the Sokoban problem in some 2D grid world and run VI on it. This method performs significantly worse than our proposed method. Concurrently with our work, Weber et al. (Weber et al. 2017) proposed a DNN architecture that combines model based planning with model free components for reinforcement learning, and demonstrated results on the Sokoban domain. In comparison, our IL approach requires significantly less training instances of the planning problem (over 3 orders of magnitude) to achieve similar performance in Sokoban.

The 'one-shot' techniques in (Duan et al. 2017), however, are complimentary to this work. The impressive Alpha-Go-Zero (Silver et al. 2017) program learned a DNN policy for Go using reinforcement learning and self play. Key to its success is the natural curriculum in self play, which allows reinforcement learning to gradually explore more complicated strategies. A similar self-play strategy was essential for Tesauro's earlier Backgammon agent (Tesauro 1995). For the goal-directed planning problems we consider here, it is not clear how to develop such a curriculum strategy, although our leapfrogging idea takes a step in that direction. Extending

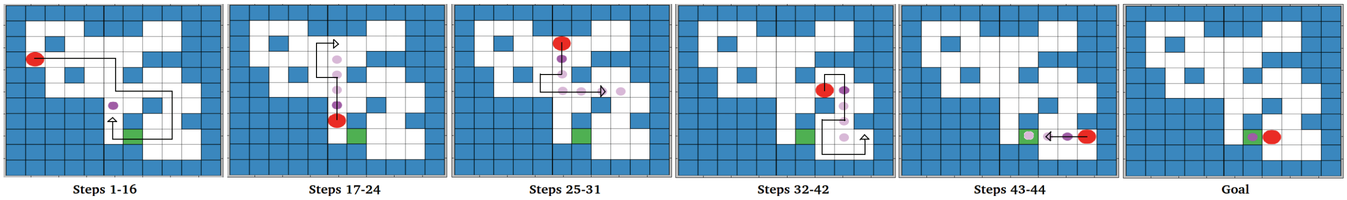


Figure 1: The Sokoban domain (best viewed in color). In Sokoban the agent (red dot) needs to push around movable objects (purple dots) between unmovable obstacles (blue squares) to a goal position (green square). In this figure we show a challenging Sokoban instance with one object. From left to right, we plot several steps in the shortest plan for this task: arrows represent the agent's path, and light purple dots show the resulting object movement. This 44 step trajectory was produced by our learned DNN policy. Note that it demonstrates reasoning about dead ends that may happen many steps after the initial state.

our work to reinforcement learning is a direction for future research.

Our approach thus offers two major advantages over prior efforts: (1) in situations where successful plan executions can be observed, e.g. by observing humans solving problems, our approach significantly reduces the effort required in designing domain representations; (2) in situations where guarantees of success are required, and domain representations are available, our approach provides an avenue for automatically generating a representation-independent heuristic function, which can be used with arbitrary guided search algorithms.

Formal Framework

We assume the reader is familiar with the formalization of deterministic, fully observable planning domains and planning problems in languages such as PDDL (Fox and Long 2003; Helmert 2009) and present the most relevant concepts here. A planning problem domain can be defined as a tuple $K = \langle \mathcal{R}, \mathcal{A} \rangle$, where $\mathcal{R}$ is a set of binary *relations*; and $\mathcal{A}$ is a set of *parameterized actions*. Each action in $\mathcal{A}$ is defined by a set of preconditions categorizing the states on which it can be applied, and the set of instantiated relations that will changed to true or false as a result of executing that action. A planning problem instance associated with a planning domain can be defined as $\Pi = \langle \mathcal{E}, s_0, G \rangle$, where $\mathcal{E}$ is a set of entities, s_0 is an initial state and G is a set of goal conditions. Relations in $\mathcal{R}$ instantiated with entities from $\mathcal{E}$ define the set of *grounded fluents*, $\mathcal{F}$. Similarly, actions in $\mathcal{A}$ instantiated with appropriately entities in $\mathcal{E}$ define the set of *grounded actions*, denoted as $\mathcal{A}[\mathcal{E}]$. The initial state, s_0, for a given planning problem is a complete truth valuation of fluents in $\mathcal{F}$; the goal condition, G, is a truth valuation of a subset of the grounded fluents in $\mathcal{F}$.

As an example, the discrete move action could be represented as follows:

$$\textit{Move}(\text{loc1}, \text{loc2}) : \begin{cases} \textit{pre} : \textit{RobotAt}(\textit{loc}1), \\ \textit{eff} : \neg \textit{RobotAt}(\text{loc1}), \textit{RobotAt}(\text{loc2}). \end{cases}$$

We introduce several additional notations to the planning problem, to make the connection with imitation learning clearer. Given a planning domain and a planning problem instance, we denote by $S = 2^{\mathcal{F}}$ the state space of the planning problem. A state $s \in S$ corresponds to the values of each fluent in $\mathcal{F}$. The task in planning is to find a sequence of grounded actions, $a_0, \dots, a_n$ – the so called *plan* – such that $a_n(\dots(a_0(s_0))\dots) \models G$.

In Sokoban, the domain represents the legal movement actions and the notion of movement on a bounded grid, a problem instance represents the exact grid layout (denoting which cell-entities are blocked), the starting locations of the objects and the agent, and the goal locations of the objects.

We denote by $o(\Pi, s)$ the *observation* for a problem instance Π when the state is s. For example, o can be an image of the current game state (Figure 1) for Sokoban. We let $\tau = \{s_0, o_0, a_0, s_1, \dots, s_g, o_g\}$ denote the state-observation-action trajectory implied by the plan. The plan length is the number of states in τ.

Our objective is to learn a generalized behavior representation that efficiently solves multiple problem instances for a domain. More precisely, given a domain K, and a problem instance Π, let $\mathcal{O}_{K,\Pi}$ be the set of possible observations of states from Π. Given a planning problem domain $K = \langle \mathcal{R}, \mathcal{A} \rangle$ we define a *generalized reactive policy (GRP)* as a function mapping observations of problem instances and states to actions: $GRP_K : \cup_\Pi \{\mathcal{O}_{K,\Pi}\} \rightarrow \cup_\Pi \{\mathcal{A}[\mathcal{E}_\Pi]\}$, where $\mathcal{E}_\Pi$ is the set of entities defined by the problem Π and the unions range over all possible problem instances associated with K. Further, GRP_K is constrained so that the observations from every problem instance are mapped to the grounded actions for that problem instance ($\forall \Pi \quad GRP_K(\mathcal{O}_{K,\Pi}) \subseteq \mathcal{A}[\mathcal{E}_\Pi]$). This effectively generalizes the concept of a policy to functions that can map states from multiple problem instances of a domain to action spaces that are legal within those instances.

Imitation Learning In imitation learning (IL), demonstrations of an expert solving a problem are given in the form of observation-action trajectories $D_{\text{imitation}} = \{o_0, a_0, o_1, \dots, o_T, a_T\}$. The goal is to find a policy – a mapping from observation to actions $a = \mu(o)$, which imitates the expert. A straightforward IL approach is *behavioral cloning* (Pomerleau 1989), in which supervised learning is used to learn μ from the data.

Learning Generalized Reactive Policies

We assume we are given a set D_{train} of N_{train} problem instances $\{\Pi_1, \dots, \Pi_{N_{\text{train}}}\}$, which will be used for learning a GRP, and a set D_{test} of N_{test} problem instances that will be used for evaluating the learned model. We also assume that the training and test problem instances are similar in some sense, so that relevant knowledge can be extracted from the training set to improve performance on the test set. Concretely, both training and test instances come from the same distribution.

Our approach consists of two stages: a data generation stage and a policy training stage.

Data generation We generate a random set of problem instances D_{train}. For each $\Pi \in D_{\text{train}}$, we run an off-the-shelf planner to generate a plan and corresponding trajectory τ, and then add the observations and actions in τ to $D_{\text{imitation}}$. In our experiments we used the Fast-Forward (FF) planner (Jörg Hoffman 2001), though any other PDDL planner can be used instead.

Policy training Given the generated data $D_{\text{imitation}}$, we use IL to learn a GRP μ. The learned policy μ maps an observation to action, and therefore can be readily deployed to any test problem in D_{test}.

One may wonder why such a naive approach would even learn to produce the complex decision making ability that is required to solve unseen instances in D_{test}. Indeed, as we show in our experiments, naive behavioral cloning with standard shallow neural networks fails on this task. One of the contributions of this work is the investigation of DNN representations that make this simple approach succeed.

Data Bootstrapping

In the IL literature (e.g., (Pomerleau 1989; Ross, Gordon, and Bagnell 2011)), the policy is typically structured as a mapping from the observation of a state to an action. However, GRPs need to consider the problem instance while generating an action to be executed since different problem instances may have different goals. Although this seems to require more data, we present an approach for "data bootstrapping" that mitigates the data requirements.

Recall that our training data $D_{\text{imitation}}$ consists of N_{train} trajectories composed of observation-action pairs. This means that the number of training samples for a policy mapping state-observations to actions is equal to the number of observation-action pairs in the training data. However, since GRPs use the goal condition in their inputs (captured by a problem instance),*any pair* of observations from successive states $(o(\Pi, s_i), o(\Pi, s_j))$ and the intermediate trajectory in an execution in D_{train} can be used as a sample for training the policy by setting s_j as a goal condition for the intermediate trajectory. Our reasoning for this data bootstrapping technique is based on the following fact:

Proposition 1. *For a planning problem Π with initial state s_0 and goal state s_g, let $\tau_{opt} = \{s_0, s_1, \ldots, s_g\}$ denote the shortest plan from s_0 to s_g. Let $\mu_{opt}(s)$ denote an optimal policy for Π in the sense that executing it from s_0 generates the shortest path τ_{opt} to s_g. Then, μ_{opt} is also optimal for a problem Π with the initial and goal states replaced with any two states $s_i, s_j \in \tau_{opt}$ such that $i < j$.*

Proposition 1 underlies classical planning methods such as triangle tables (Fikes, Hart, and Nilsson 1972). Here, we exploit it to design our DNN to take as input *both* the *current observation* and a *goal observation*. For a given trajectory of length T, the bootstrap can potentially increase the number of training samples from T to $(T-1)^2/2$. In practice, for each trajectory $\tau \in D_{\text{imitation}}$, we uniformly sample $n_{\text{bootstrap}}$ pairs of observations from τ. In each pair, the first observation is treated as the current observation, while the last observation is treated as the goal observation[1]. This results in $n_{\text{bootstrap}}+T$ training samples for each trajectory τ, which are added to a bootstrap training set $D_{\text{bootstrap}}$ to be used instead of $D_{\text{imitation}}$ for training the policy. [2]

Network Structure

We propose a general structure for a convolutional network that can learn a GRP.

Our network is depicted in Figure 2. The current state and goal state observations are passed through several layers of convolution which are shared between the action prediction network and the plan length prediction network. There are also skip connections from the input layer to to every convolution layer.

The shared representation is motivated by the fact that both the actions and the overall plan length are integral parts of a plan. Having knowledge of the actions makes it easy to determine plan length and vice versa, knowledge about the plan length can act as a template for determining the actions. The skip connections are motivated by the fact that several planning algorithms can be seen as applying a repeated computation, based on the planning domain, to a latent variable. For example, greedy search expands the current node based on the possible next states, which are encoded in the domain; value iteration is a repeated modification of the value given the reward and state transitions, which are also encoded in the domain. Since the network receives no other knowledge about the domain, other than what's present in the observation, we hypothesize that feeding the observation to every conv-net layer can facilitate the learning of similar planning computations. We note that in value iteration networks (Tamar et al. 2016), similar skip connections were used in an explicit neural network implementation of value iteration.

For planning in graph domains, we propose to use graph convolutions, similar to the work of (Dai et al. 2017). The graph convolution can be seen as a generalization of an image convolution, where an image is simply a grid graph. Each node in the graph is represented by a feature vector, and linear operations are performed between a node and its neighbors, followed by a nonlinear activation. A detailed description is provided in the supplementary material. For the TSP problem with n nodes, we map a partial Hamiltonian path P of the graph to a feature representation as follows. For each node, the features are represented as a 3-dimensional binary vector.

[1] In our experiments, we used the FF planner, which does not necessarily produce shortest plans. However, Proposition 1 can be extended to satisficing plans.

[2] Note that for the Sokoban domain, goal observations in the test set (i.e., real goals) do not contain the robot position, while the goal observations in the bootstrap training set include the robot position. However, this inconsistency had no effect in practice, which we verified by explicitly removing the robot from the observation.

The first element is 1 if the node has been visited in P, the second element is 1 if it is the current location of the agent, and the third element is 1 if the node is the terminal node. For a Hamiltonian cycle the terminal node is the start node. The state is then represented as a collection of feature vectors, one for each node. In the TSP every Hamiltonian cycle is of length n, so predicting the plan length in this case is trivial, as we encode the number of visited cities in the feature matrix. Therefore, we omit the plan-length prediction part of the network.

Generalization to Different Problem Sizes

A primary challenge in learning for planning is finding representations that can generalize across different problem sizes. For example, we expect that a good policy for Sokoban should work well on the instances it was trained on, 9×9 domains for example, as well as on larger instances, such as 12×12 domains. A convolution-based architecture naturally addresses this challenge.

However, while the convolution layers can be applied to any image/graph size, the number of inputs to the fully connected layer is strictly tied to the problem size. This means that the network architecture described above is fixed to a particular grid dimension. To remove this dependency, we employ a trick used in fully convolutional networks (Long, Shelhamer, and Darrell 2015), and keep only a $k \times k$ window of the last convolution layer, centered around the current agent position. This modification makes our DNN applicable to any grid dimension. Note that since the window is applied *after* the convolution layers, the receptive field can be much larger than $k \times k$. In particular, a value of $k = 1$ worked well in our experiments. For the graph architectures, a similar trick is applied, where the decision at a particular node is a function of the convolution result of its neighbors, and the same convolution weights are used across different graph sizes.

Experiments

Here we report our experiments on learning for planning with DNNs. Our focus is on the following questions:

1. What makes a good DNN architecture for learning a GRP?
2. Can a useful planning heuristic be extracted from the GRP?

The first question aims to show that recent developments in the representation learning community, such as deep convolutional architectures, can be beneficial for planning. The second question has immediate practical value – a good heuristic can decrease planning costs. However, it also investigates a deeper premise. If a useful heuristic can indeed be extracted from the GRP, it means that the GRP has learned some underlying structure in the problem. In the domains we consider, such structure is hard to encode manually, suggesting that the data-driven DNN approach can be promising.

To investigate these questions, we selected two test domains representative of very different classes of planning problems. We used the *Sokoban* domain to represent problems where plan execution can be captured as a set of images, and the goal takes the form of achieving a state property (objects at their target locations). We used the *traveling salesperson problem* as an exemplar for problems where plan execution is not easy to capture as a set of images and the goal features a temporal property.

Sokoban For Sokoban, we consider two difficulty levels: moving a single object as described in Figure 1, and a harder task of moving two objects. We generated training data using a Sokoban random level generator[3].

For imitation learning, we represent the policy with the DNNs as described in Network Structure section and optimize using Adam (Kingma and Ba 2014) (step size 0.001). When training with data bootstrapping, we selected $n_{\text{bootstrap}} = T$ for generating $D_{\text{bootstrap}}$. Unless stated otherwise, the training set used in all Sokoban experiments was comprised of 45k observation-action trajectories from 9k different obstacle configurations.

To evaluate policy performance on the Sokoban domain we use execution success rate. Starting from the initial state, we execute the learned policy deterministically and track whether or not the goal state is reached. We evaluate performance both on test domains of the same size the GRPs were trained on, 9×9 grids, and also on larger problems. We explicitly verified that *none of the test domains appeared in the training set*.

Videos of executions of our learned GRPs for Sokoban are available at goo.gl/Hpy4e3.

TSP For TSP, we consider two different graph distributions. The first is the space of complete graphs with edge weights sampled uniformly in $[0, 1]$. The second, which we term *chord graphs*, is generated by first creating an n-node graph in the form of a cycle, and then adding $2n$ undirected chords between randomly chosen pairs of nodes, with a uniformly sampled weight in $[0, 1]$. The resulting graphs are guaranteed to contain Hamiltonian cycles. However, in contrast to the complete graphs, finding such a Hamiltonian cycle is not trivial. Our results for the chord graphs are similar to the complete graphs, and for space constraints, we present them in the supplementary material. Training data was generated using the TSP solver in Google Optimization Tools[4].

As before, we train the DNN using Adam. We found it sufficient to use only 1k observation-action trajectories for our TSP domain. The metric used is average relative cost[5], defined as the ratio between the cycle cost of the learned policy and the Google solver, averaged over all initial nodes in each test domain. We also compare the DNN policy against a greedy policy which always picks the lowest-cost edge

[3] The Sokoban data-set from the learning for planning competition contains only 60 training domains, which is not enough to train a DNN. Our generator works as follows: we assume the room dimensions are a multiple of 3 and partition the grid into 3x3 blocks. Each block is filled with a randomly selected and randomly rotated pattern from a predefined set of 17 different patterns. To make sure the generated levels are not too easy and not impossible, we discard the ones containing open areas greater than 3x4 and discard the ones with disconnected floor tiles. For more details we refer the reader to Taylor et al. (Taylor and Parberry 2011).

[4] https://developers.google.com/optimization

[5] For the complete graphs, all policies always succeeded in finding a Hamiltonian cycle. For the chord graphs, we report success rates in the supplementary material.

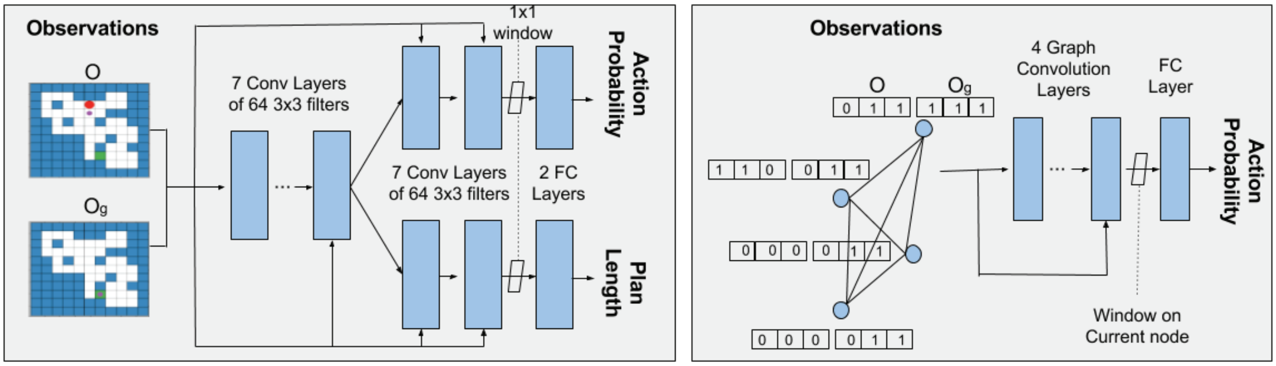


Figure 2: Network architecture. The architecture on the left is used for Sokoban, while the one on the right is used for the TSP. A pair of current and goal observations are passed in to a shared conv-net. This shared representation is input to an action prediction conv-net and a plan length prediction conv-net. Skip connections from the input observations to all conv-layers are added. For the TSP network, we omitted the plan length prediction, as the features directly encode the number of nodes visited, making the prediction trivial. All activation functions are ReLU's and the final one is a SoftMax. In both architectures, after the last convolution layer, we apply a $k \times k$ window around the agents location to ensure a constant size feature vector is passed to the fully connected layers. This effectively decouples the architecture from the problem size and allows the receptive field to be greater than the $k \times k$ window.

leading to an unvisited node.

As in the Sokoban domain, we evaluate performance on test domains with graphs of the same size as the training set, 4 node graphs, and on larger graphs with up-to 11 nodes.

Evaluation of Learned GRPs

Here we evaluate performance of the learned GRPs on previously unseen test problems. Our results suggest that the GRP can learn a well-performing planning-like policy for challenging problems. In the Sokoban domain, on 9×9 grids, the learned GRP in the best performing architecture (14 layers, with bootstrapping and a shared representation) can solve one-object Sokoban with 97% success rate, and two-object Sokoban with 87% success rate. Figure 1 shows a trajectory that the policy predicted in a challenging one-object domain from the test set. Two-object trajectories are difficult to illustrate using images; we provide a video demonstration at goo.gl/Hpy4e3. We observed that the GRP effectively learned to select actions that avoid dead ends far in the future, as Figure 1 demonstrates. The most common failure mode is due to cycles in the policy, and is a consequence of using a deterministic policy. Due to space constraints, further analysis of failure modes is given in the supplementary material. The learned GRP can thus be used to solve new planning problem instances with a high chance of success. In domains where simulators are available, a planner can be used as a fallback if the policy fails in simulation.

Figure **??** shows the performance of the GRP policy on complete graphs of sizes $4 - 11$, when trained on graphs of the same size (respectively). For both the GRP and the greedy policy, the cost increases approximately linearly with the graph size. For the greedy policy, the rate of cost increase is roughly twice the rate for the GRP, showing that the GRP learned to perform some type of lookahead planning.

Investigation of Network Structure

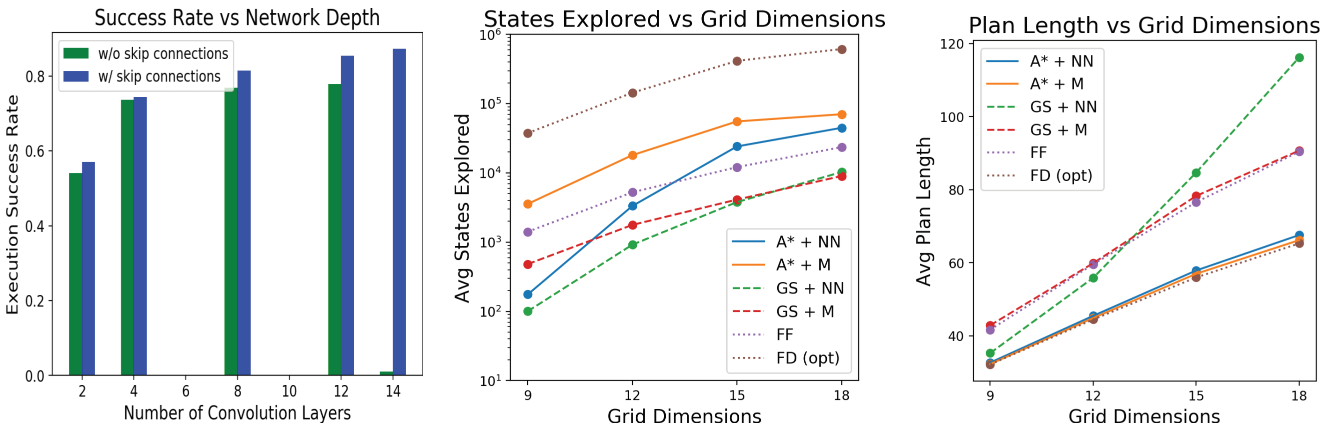
We performed ablation experiments to tease out the important ingredients for a successful GRP. Our results suggest that deeper networks improve performance.

In Figure 3a we plot execution success rate on two-object Sokoban, for different network depths, and with or without skip connections. The results show that deeper networks perform better, with skip connections resulting in a consistent advantage. In the supplementary material we show that a deep network significantly outperformed a shallow network with the same number of parameters, further establishing this claim. The improved results for the deeper networks suggest that for learning GRP's – **the deeper the network the better**. We note a related observation in the context of a DNN representation of the value iteration planning algorithm in (Tamar et al. 2016). However, in our experiments the performance levels off after 14 layers. We attribute this to the general difficulty of training deep DNNs due to gradient propagation, as evident in the failure of training the 14 layer architecture without skip connections, Figure 3a.

We also investigated the benefit of having a shared representation for both action and plan length prediction, compared to predicting each with a separate network. The ablation results are presented in Table 1. Interestingly, the plan length prediction improves the accuracy of the action prediction.

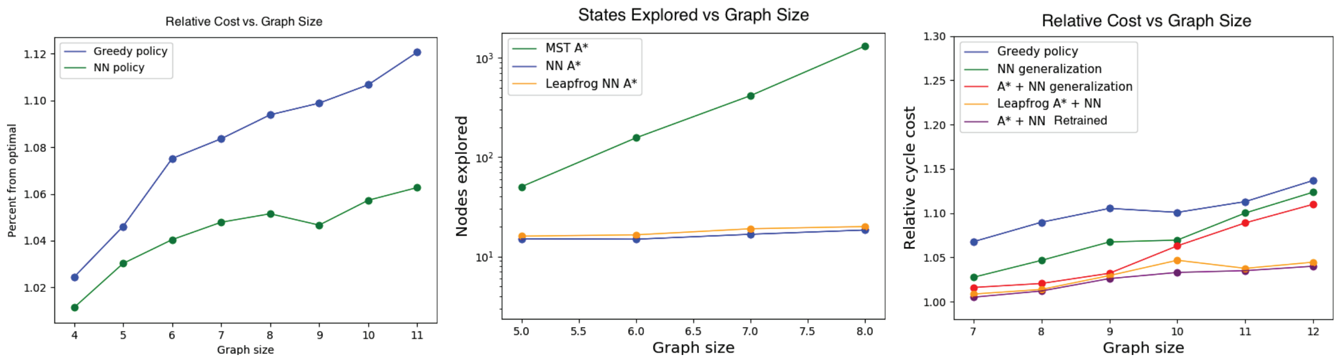
GRP as a Heuristic Generator

We now show that the learned GRPs can be used to extract *representation independent heuristics* for use with arbitrary guided search algorithms. To our knowledge, there are no other approaches for computing such heuristics without using hand-curated domain vocabularies or features for learning and/or expressing them. However, to evaluate the quality of our learned heuristics, we compared them with a few well-known heuristics that are either handcrafted or com-



(a) (b) (c)

Figure 3: Sokoban results. (a) Investigating DNN depth and skip connections. We plot the success rate for deterministic execution in two-object Sokoban. Deeper networks show improved success rates and skip connections improve performance consistently. We were unable to successfully train a 14 layer deep network without skip connections. (b,c) Performance of learned heuristic. The GRP was trained only on 9x9 instances, and evaluated (as a heuristic, see text for more details) on larger instances. (b) shows number of states explored (i.e., planning speed) and (c) shows plan length (i.e., planning quality). A* with the learned heuristic produced nearly optimal plans with an order of magnitude reduction in the number of states explored.



(a) (b) (c)

Figure 4: TSP results. (a) Performance (average relative cost; see text for details) for GRPs trained and tested on problems of sizes $4-11$, respectively. We compare the GRP with a greedy policy. (b,c) Performance of learned heuristic. The GRP was trained on 4-node graphs, and evaluated (as a heuristic, see text for more details) on larger instances. (b) shows number of states explored (i.e., planning speed). We compare with the minimum spanning tree heuristic, which is admissible for TSP. (c) shows average relative cost (i.e., planning quality) compared to plans from the Google solver. Note that up to a graph of size 9, the performance of A* with GRP heuristic (labeled A*+NN generalization) was within 5% of optimal, while requiring orders of magnitude less computation than the MST heuristic. We also present results for the leapfrogging algorithm (see text for details), and additionally compare to a baseline of retraining the GRP with optimal data for each graph size. Note that the leapfrogging results are very close to the results obtained with retraining, although optimal data was only given for the smallest graph size. This shows that the GRP heuristic can be used for generating reliable training data for domains of larger size than trained on.

puted using handcrafted representations. We found that the representation-independent GRP heuristic was competitive, and remains effective on larger problems than the GRP was trained on. For the Sokoban domain, the plan-length prediction can be directly used as a heuristic function. This approach can be used for state-property based goals in problems where execution can be captured using images. For the TSP domain, we used a heuristic that is inversely proportional to the probability of selecting the next node to visit, as the number of steps required to create a complete cycle is not discriminative. Full details are given in the supplementary material.

We investigated using the GRP as a heuristic in greedy search and A* search (Hart, Nilsson, and Raphael 1968). We use two performance measures: the number of states explored during search and the length of the computed plan. The first measure corresponds to planning speed since evaluating less nodes translates to faster planning. The second measure rep-

	w/ bootstrap	w/o bootstrap	
Predict plan length	2.211	2.481	ℓ_1 norm
Predict plan length & actions	**2.205**	2.319	ℓ_1 norm
	0.844	0.818	Succ Rate
Predict actions	0.814	0.814	Succ Rate

Table 1: Benefits of bootstrapping and having a shared representation. To evaluate accuracy of the plan length prediction, we measure the average ℓ_1 loss (absolute difference). To evaluate action prediction we measure the success rate on execution. Best performance was obtained with using bootstrapping and the shared representation. For this experiment the training set contained 25k observation-action trajectories.

resents plan quality.

Sokoban We compare performance in Sokoban to the Manhattan heuristic[6] in Figure 3b. In the same figure we evaluate generalization of the learned heuristic to larger, never before seen, instances as well as the performance of two state-of-the-art planners: Fast Forward (FF, (Jörg Hoffman 2001)) and Fast Downward (FD, (Helmert 2006))[7]. The GRP was trained on 9×9 domains, and evaluated on new problem instances of similar size or larger. During training, we chose the window size $k = 1$ to influence learning a problem-instance-size-invariant policy. As seen in Figure 3b the learned GRP heuristic *significantly outperforms the Manhattan heuristic* in both greedy search and A* search, on the 9x9 problems. As the size of the test problems increases, the learned heuristic shines when used in conjunction with A*, consistently expanding fewer nodes than the Manhattan heuristic. Note that even though the GRP heuristic is not guaranteed to be admissible, when used with A*, the plan quality is very close to optimal, while exploring an order of magnitude less nodes than the conventional alternatives.

TSP We trained the GRP on 6-node complete graphs and evaluated the GRP, used either directly as a policy or as a heuristic within A^*, on graphs of larger size. Figure 4(b-c) shows generalization performance of the GRP, both in terms of planning speed (number of nodes explored) and in terms of plan quality (average relative cost). We compare both to a greedy policy, and to A^* with the minimum spanning tree (MST) heuristic. Note that the GRP heuristic is significantly more efficient than MST, while not losing much in terms of plan quality, especially when compared to the greedy policy.

Leap-Frogging Algorithm

The effective generalization of the GRP heuristic to larger problem sizes motivates a novel algorithmic idea for learning to plan on iteratively increasing problem sizes, which we term *leap-frogging*. The idea is that, we can use a 'general and optimal' planner, such as FD, to generate data for a small domain, of size d. We then train a GRP using this data, and use the resulting GRP heuristic in A^* to *quickly* solve planning problems from a larger domain $d' > d$. These solutions can then be used as new data for training another GRP on the domain size d'. Thus, we can iteratively apply this procedure to solve problems of larger and larger sizes, while only requiring the slow 'general' planner to be applied in the smallest domain size.

In Figure 4c we demonstrate this idea in the TSP domain. We used the solver to generate training data for a graph with 4 nodes. We then evaluate the GRP heuristic trained using leapfrogging on larger domains, and compare with a GRP heuristic that was only trained on the 4-node graph. Note that we significantly improve upon the standard GRP heuristic, while using the same initial optimal data obtained from the slow Google solver. We also compare with a GRP heuristic that was re-trained with optimal data for each graph size. Interestingly, this heuristic performed only slightly better than the GRP trained using leap-frogging, showing that the generalization of the GRP heuristic is effective enough to produce reliable new training data.

Conclusion

We presented a new approach in learning for planning, based on imitation learning from execution traces of a planner. We used deep convolutional neural networks for learning a generalized policy, and proposed several network designs that improve learning performance in this setting, and are capable of generalization across problem sizes. In addition, we showed that our networks can be used to extract a heuristic for off-the-shelf planners, which led to significant improvements over standard heuristics that do not leverage learning.

Our results on the challenging Sokoban domain suggest that DNNs have the capability to extract powerful features from observations, and the potential to learn the type of 'visual thinking' that makes some planning problems easy for humans but very hard for automatic planners. The leapfrogging results, suggest a new approach for planning – when facing a large and difficult problem, first solve simpler instances of the problem and learn a DNN heuristic that aids search algorithms in solving larger instances. This heuristic can be used to generate data for training a new DNN heuristic for larger instances, and so on. Our preliminary results suggest this approach to be promising.

There is still much to explore in employing deep networks for planning. While representations for images based on deep conv-nets have become standard, representations for other modalities such as graphs and logical expressions are an active research area (Dai et al. 2017; Kansky et al. 2017). We believe that the results presented here will motivate future research in representation learning for planning.

References

Bylander, T. 1994. The computational complexity of propositional strips planning. *Artificial Intelligence* 69(1-2):165–204.

[6] The Manhattan heuristic is only admissible in one-object Sokoban. We tried Euclidean distance and Hamiltonian distance. However, Manhattan distance had the best trade-off between performance and computation time.

[7] FD uses an anytime algorithm, so we constrained the planning time to be no more than 5 minutes per instance. For the problem instances we evaluated, FD always found the optimal solution.

Dai, H.; Khalil, E. B.; Zhang, Y.; Dilkina, B.; and Song, L. 2017. Learning combinatorial optimization algorithms over graphs. *arXiv preprint arXiv:1704.01665*.

Duan, Y.; Andrychowicz, M.; Stadie, B.; Ho, J.; Schneider, J.; Sutskever, I.; Abbeel, P.; and Zaremba, W. 2017. One-shot imitation learning. *arXiv preprint arXiv:1703.07326*.

Ernandes, M., and Gori, M. 2004. Likely-admissible and sub-symbolic heuristics. In *Proceedings of the 16th European Conference on Artificial Intelligence*, 613–617. IOS Press.

Fern, A.; Khardon, R.; and Tadepalli, P. 2011. The first learning track of the international planning competition. *Machine Learning* 84(1):81–107.

Fikes, R. E.; Hart, P. E.; and Nilsson, N. J. 1972. Learning and executing generalized robot plans. *Artificial Intelligence* 3:251 – 288.

Fox, M., and Long, D. 2003. PDDL2. 1: An extension to PDDL for expressing temporal planning domains. *J. Artif. Intell. Res.(JAIR)* 20:61–124.

Hart, P. E.; Nilsson, N. J.; and Raphael, B. 1968. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics* 4(2):100–107.

Helmert, M. 2006. The fast downward planning system. *Journal of Artificial Intelligence (JAIR)* 26:191–246.

Helmert, M. 2009. Concise finite-domain representations for pddl planning tasks. *Artificial Intelligence* 173(5):503 – 535.

Hu, Y., and De Giacomo, G. 2011. Generalized planning: Synthesizing plans that work for multiple environments. In *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*.

Jörg Hoffman. 2001. FF: The fast-forward planning system. *AI Magazine* 22:57–62.

Kambhampati, S., and Kedar, S. 1994. A unified framework for explanation-based generalization of partially ordered and partially instantiated plans. *Artificial Intelligence* 67(1):29–70.

Kansky, K.; Silver, T.; Mély, D. A.; Eldawy, M.; Lázaro-Gredilla, M.; Lou, X.; Dorfman, N.; Sidor, S.; Phoenix, S.; and George, D. 2017. Schema networks: Zero-shot transfer with a generative causal model of intuitive physics. *arXiv preprint arXiv:1706.04317*.

Khardon, R. 1999. Learning action strategies for planning domains. *Artificial Intelligence* 113(1):125 – 148.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440.

Martín, M., and Geffner, H. 2004. Learning generalized policies from planning examples using concept languages. *Applied Intelligence* 20(1):9–19.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Mülling, K.; Kober, J.; Kroemer, O.; and Peters, J. 2013. Learning to select and generalize striking movements in robot table tennis. *The International Journal of Robotics Research* 32(3):263–279.

Nair, A.; Chen, D.; Agrawal, P.; Isola, P.; Abbeel, P.; Malik, J.; and Levine, S. 2017. Combining self-supervised learning and imitation for vision-based rope manipulation. *arXiv preprint arXiv:1703.02018*.

Pfeiffer, M.; Schaeuble, M.; Nieto, J.; Siegwart, R.; and Cadena, C. 2016. From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. *arXiv preprint arXiv:1609.07910*.

Pomerleau, D. A. 1989. Alvinn: An autonomous land vehicle in a neural network. In *Advances in Neural Information Processing Systems*, 305–313.

Ross, S.; Gordon, G. J.; and Bagnell, D. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*.

Scala, E.; Torasso, P.; et al. 2015. Deordering and numeric macro actions for plan repair. In *IJCAI*, 1673–1681.

Shavlik, J. W. 1989. Acquiring recursive concepts with explanation-based learning. In *IJCAI*, 688–693.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *Nature* 550(7676):354–359.

Spalzzi, L. 2001. A survey on case-based planning. *Artificial Intelligence Review* 16(1):3–36.

Srivastava, S.; Immerman, N.; and Zilberstein, S. 2011. A new representation and associated algorithms for generalized planning. *Artificial Intelligence* 175(2):615–647.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Tamar, A.; Levine, S.; Abbeel, P.; WU, Y.; and Thomas, G. 2016. Value iteration networks. In *Advances in Neural Information Processing Systems*, 2146–2154.

Taylor, J., and Parberry, I. 2011. Procedural generation of sokoban levels. In *Proceedings of the International North American Conference on Intelligent Games and Simulation*.

Tesauro, G. 1995. Temporal difference learning and td-gammon. *Communications of the ACM* 38(3):58–68.

Weber, T.; Racanière, S.; Reichert, D. P.; Buesing, L.; Guez, A.; Rezende, D. J.; Badia, A. P.; Vinyals, O.; Heess, N.; Li, Y.; et al. 2017. Imagination-augmented agents for deep reinforcement learning. *arXiv preprint arXiv:1707.06203*.

Yoon, S.; Fern, A.; and Givan, R. 2002. Inductive policy selection for first-order MDPs. In *Proceedings of the Eigh-*

teenth conference on Uncertainty in artificial intelligence, 568–576. Morgan Kaufmann Publishers Inc.

Yoon, S.; Fern, A.; and Givan, R. 2008. Learning control knowledge for forward search planning. *Journal of Machine Learning Research* 9(Apr):683–718.

Appendix

Graph Convolution Network

Consider a graph $\mathcal{G} = (V, \mathcal{E})$ with adjacency matrix A where V has N nodes and $\mathcal{E}$ is the weighted edge set with weight matrix W. Suppose that each node $v \in V$ has a corresponding feature $x_v \in \mathbb{R}^m$ and consider a parametric function $f_\theta : \mathbb{R}^{2m} \to \mathbb{R}^m$ parameterized by $\theta \in \mathbb{R}^f$. Let $\mathcal{N}_i : V \to 2^V$ denote a function mapping a vertex to its ith degree neighborhood. The propagation rule is given by the following equation

$$H_v = \sigma\left(\sum_{u\in\mathcal{N}(v)} A_{uv} f_\theta(x_u, x_v)\right) \tag{1}$$

where σ is the ReLU function. Consider a graph $\mathcal{G}$ of size n, with each vertex having feature vector of size C encoded in the feature matrix $X \in \mathbb{R}^{n\times C}$. In the TSP experiments, we use the propagation rule where the ij entry of the next layer is given by

$$H_{ij} = \sigma\left(\sum_{s\in\mathcal{N}(i)} A_{si}[x_s, x_i, W_{si}]^T \Theta_j + b_j\right) \tag{2}$$

Here, W is the weight matrix of $\mathcal{G}$, A is the adjacency matrix, and $\Theta \in \mathbb{R}^{(2C+1)\times C'}$ is the matrix of weights that we learn and $b \in \mathcal{R}^{C'}$ is a learned bias vector. Θ_j is the jth column of Θ.

In the networks we used for the TSP domain, the initial feature vector is of size $C = 6$. We then applied 4 convolution layers of size $C = 26$. We then applied a convolution of size $C = 1$, corresponding to a fully connected layer. Thus, $j = 1$ in H_{ij} for all i in the last convolution layer.

The final layer of the network is a softmax over H_{i1}, and we select the node i with the highest score that is also connected to the current node.

Relation to Image Convolution In the next proposition we show that this graph-based propagation rule can be seen as a generalization of a standard 2-D convolution, when applied to images (grid graphs). Namely, we show that there exists features for a grid graph and parameters Θ for which the above propagation rule reduces to a standard 2-D convolution.

Proposition 2. *When $\mathcal{G}$ is a grid graph, for a particular choice of f_θ the above propagation rule reduces to the traditional convolutional network. In particular, for a filter of size n, choosing f_θ as a polynomial of degree $2(N-1)$ and $\theta \in \mathbb{R}^{N^2}$ works.*

Proof. For each node v, consider its representation as $v = (v_x, v_y)$ where (v_x, v_y) are the grid coordinates of the vertex.

Num Params	Deep-8	Wide-2	Wide-1	
556288	0.068	0.092	0.129	error rate
	0.83	0.62	0.38	succ rate

Table 2: Comparison of deep vs. shallow networks. The deep network has 8 convolution layers with 64 filter per layer. The shallow networks contain 2 and 1 layers respectively with 256 and 512 filters per layer respectively. Clearly, deeper networks outperform shallow networks while containing an equal number of parameters.

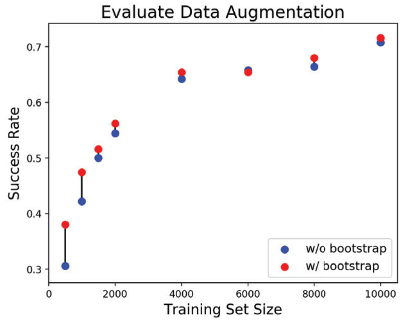


Figure 5: This shows the affect of data bootstrapping on the performance of two-object Sokoban, as a function of the dataset size. Smaller datasets benefit more from data augmentation.

Let $a := \frac{n-1}{2}$. We first transform the coordinates to center them around v by transforming $u \to (u_x - v_x, u_y - v_y)$ so that u lies in the set $[-a, a] \times [-a, a]$.

We wish to design a polynomial g that takes the value $\theta_{i,j}$ at location (i, j). We show that it is possible to do with a degree $2(n-1)$ polynomial by construction. The polynomial g is given by

$$g(x, y) := \sum_{i=-a}^{a}\sum_{j=-a}^{a} \theta_{i,j} \prod_{s=-a, s\neq i}^{a} (s + y) \prod_{t=-a, t\neq j}^{a} (t + x) \tag{3}$$

To see why this is correct, note that for any $(s, t) \in [-a, a] \times [-a, a]$ there is exactly one polynomial inside the summands that does not have either of the terms $(i + u_y)$ or $(j + u_x)$ appearing in its factorization. Indeed, by construction this term is the polynomial corresponding to $\theta_{i,j}$ so that $g(i, j) = C\theta_{i,j}$ for some constant C.

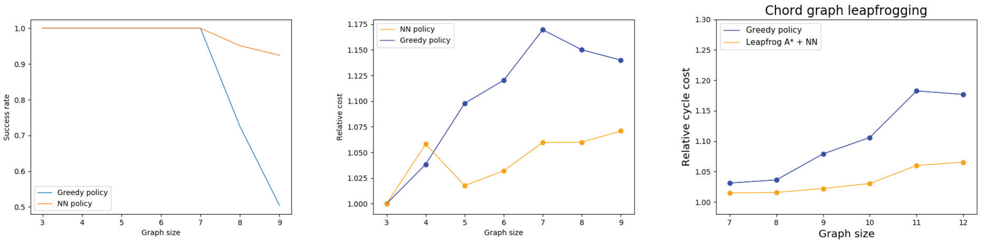
The polynomial inside the summands is of degree $(n-1) + (n-1) = 2(n-1)$, so g is of degree $2(n-1)$. Letting p_u denote th pixel value at node u, setting

$$f_\theta(x_u, x_v) := p_u g(x_u - x_v) \tag{4}$$

completes the proof. □

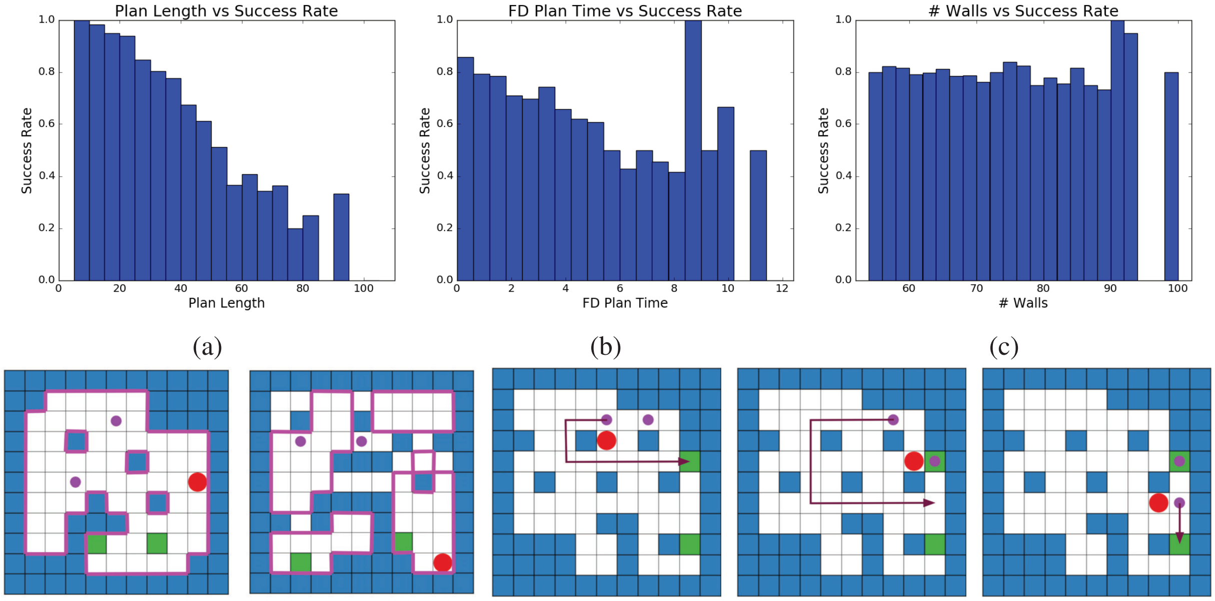
TSP domain heuristic

We can use the graph convolution network as a heuristic inside A-star search. Given a feature encoding of a partial cycle P, we can compute the probability p_i of moving to



(a) (b) (c)

Figure 6: Chord-graph TSP results. (a) Success rate of neural network policy on chord graphs of size $3 - 9$, respectively. Note that the agent is only allowed to visit each node once, so the agent may visit a node with no un-visited neighbors which is a dead end. We also show the success rate of the greedy policy. (b) Performance of neural network policy on chord graphs of size 3-9. (c) Leapfrogging algorithm results on chord graphs of size 7-12. We compare to a baseline greedy policy



(a) (b) (c)

(d) (e) (f) (g) (h)

Figure 7: Analysis of Failure Modes. (a-c): Success rate vs features of the domain. Plan length (a) seems to be the main factor in determining success rate. Longer plans fail more often. While there is some relationship between planning time and success rate (b), planning time is not always an accurate indicator, as explained in (d,e). The number of walls (c) does not affect success rate. (d,e): Domains containing large open rooms results in a high branching factor and thus produce the illusion of difficulty while still having a simple underlying policy. The domain in (d) took FD significantly longer time to solve, 8.6 seconds compared to 1.6 seconds for the domain in (e), although it has a shorter optimal solution, 51 steps compared to 65 steps. This is since the domain in (e) can be broken up into small regions which are all connected by hallways, a configuration that reduces the branching factor and thus the overall planning speed. (f-h): Demonstration of the 2nd failure mode in Section . From the start state, the policy moves the first object using the path shown in (f). It proceeds to move the next object using the path in (g). As the game state approaches (h) it becomes clear that the current domain is no longer solvable. The lower object needs to be pushed down but is blocked by the upper object, which can no longer be moved out of the way. In order to solve this level, the first object must ether be moved to the bottom goal or must be moved after the second object has been placed at the bottom goal. Both solutions require a look-ahead consisting of 20+ steps.

any node i. We then use the quantity $(N - v)(1 - p_i)/2$ as the heuristic, where N is the total number of nodes and v is the number of visited nodes in the current partial path. Multiplying by $(N - v)/2$ puts the output of the heuristic on the same scale as the current cost of the partial path.

Deep VS Shallow Networks

Here we present another experiment to further establish the claim that the depth of the network improves performance and not necessarily the number of parameters in the network. In Table 2 we compare deep networks against shallow networks containing the same number of parameters. Note that we evaluate based on two different metrics. The first metric is classification error on the next action, which shows whether or not the action matches what the planner would have done. The second metrics is execution success rate, as defined above.

Evaluation of Bootstrap Performance

We briefly summarize the evaluation of data bootstrapping in the Sokoban domain. Table 1 shows the success rate and plan length prediction error for architectures with and without the bootstrapping. As can be observed, the bootstrapping resulted in better use of the data, and led to improved results.

While investigating the performance of data bootstrapping with respect to training set size, we observed that a non-uniform sampling performed better on smaller datasets. For each $\tau \in D_{\text{imitation}}$, we sampled an observation $\hat{o}$ from a distribution that is linearly increasing in time, such that observations near the goal have higher probability. The performance of this bootstrapping strategy is shown in Figure 5. As should be expected, performance improvement due to data augmentation is more significant for smaller data sets.

Analysis of Failure Modes

While investigating the failure modes of the learned GRP in the Sokoban domain, we noticed that there were two primary failure modes. The first failure mode is due to cycles in the policy, and is a consequence of using a deterministic policy. For example, when the agent is between two objects a deterministic policy may oscillate, moving back and fourth between the two. We found that a stochastic policy significantly reduces this type of failure. However, stochastic policies have some non-zero probability of choosing actions that lead to a dead end (e.g., pushing the box directly up against a wall), which can lead to different failures. The second failure mode was the inability of our policy to foresee long term dependencies between the two objects. An example of such a case is shown in Figure 7 (f-h), where deciding which object to move first requires a look-ahead of more than 20 steps. A possible explanation for this failure is that such scenarios are not frequent in the training data. This is less a limitation of our approach and more a limitation of the neural network, more specifically the depth of the neural network.

Additionally, we investigated whether the failure cases can be related to specific features in the task. Specifically, we considered the task plan length (computed using FD), the number of walls in the domain, and the planning time with the FD planner (results are similar with other planners). Intuitively, these features are expected to correlate with the difficulty of the task. In Figure 7 (a-c) we plot the success rate vs. the features described above. As expected, success rate decreases with plan length. Interestingly, however, several domains that required a long time for FD were ‘easy’ for the learned policy, and had a high success rate. Further investigation revealed that these domains had large open areas, which are ‘hard’ for planners to solve due to a large branching factor, but admit a simple policy. An example of one such domain is shown in Figure 7 (d-e). We also note that the number of walls had no visible effect on success rate – it is the configuration of the walls that matters, and not their quantity.

Constraint-Based Online Transformation of Abstract Plans into Executable Robot Actions

**Till Hofmann,[1] Victor Mataré,[2] Stefan Schiffer,[1,2]
Alexander Ferrein,[2] Gerhard Lakemeyer[1]**
[1]Knowledge-Based Systems Group,
RWTH Aachen University, 52056 Aachen, Germany
[2]Mobile Autonomous Systems and Cognitive Robotics,
FH Aachen University of Applied Sciences, 52066 Aachen, Germany

Abstract

In this paper, we are concerned with making the execution of abstract action plans for robotic agents more robust. To this end, we propose to model the internals of a robot system and its ties to the actions that the robot can perform. Based on these models, we propose an online transformation of an abstract plan into executable actions conforming with system specifics. With our framework, we aim to achieve two goals. First, modeling the system internals is beneficial in its own right in order to achieve long term autonomy, system transparency, and comprehensibility. Second, separating the system details from determining the course of action on an abstract level leverages the use of planning for actual robotic systems.

Introduction

Despite promising advances in planning systems, they see surprisingly little use in actual robotics environments. We believe this is because solving a planning task by itself is not sufficient to accomplish high-level behavior control of a robotic system. For one, the robot's platform (i.e., its hardware and low-level software components) often requires additional constraints that are ignored during planning, e.g., a domestic service robot participating in RoboCup@Home (Wisspeintner et al. 2009) must calibrate its arm before performing any manipulation tasks. During planning, we do not want to plan for all the requirements of the underlying platform, as this would increase the problem size significantly and would make it infeasible in practice. However, ignoring those constraints at the behavior level and dealing with them at the lower levels is often impossible, because platform constraints may entail changes to the action plan.

Another reason for such a separation of high-level behavior and low-level platform is a design problem: When modelling the domain, an agent programmer usually does not want to deal with the robot platform. On the other hand, a platform designer should not need to consider and adapt the high-level behavior when modifying the platform. Also, a robot often has to deal with failed actions, unexpected changes, and exogenous events. Thus, a considerable amount of monitoring is required when executing a high-level plan on a robot.

For these reasons, we propose a framework that allows the modelling of the robot platform and its constraints independent of the behavioral component. While designing the platform, the user designs a self model of the robot and defines all the constraints of the platform. The world model of the agent can be designed without taking low-level constraints into account. During execution, the abstract action plan is transformed into a concrete executable plan that satisfies the constraints of the lower levels.

To actually achieve a separation between the problem domain and platform-related execution concerns, the platform needs a certain degree of "self-awareness" in terms of its components, their capabilities, their states and their interdependencies. Our goal in this paper is to sketch out requirements for a logically founded constraint language that can be used by platform experts to explicitly model component state transitions, dependencies among them, error conditions and possible recovery strategies, including the potential need for human assistance. The result is an agent system capable of self-maintenance by generating platform-specific monitoring and recovery strategies from the platform model and a platform-independent action plan. This eliminates much of the expert intervention that is required to keep robots running in dynamic domains, while providing a generic framework that helps in decoupling strategic decision-making from any platform details.

Foundations & Related Work

Especially the research into planning systems that is focused on temporal coordination of (concurrent) actions is of particular interest to our endeavour (Tsamardinos, Muscettola, and Morris 1998; Jónsson et al. 2000; Kim, Williams, and Abramson 2001; Lemai and Ingrand 2004). In theory, it would allow generalizing both the domain logic and the platform details as a temporal planning problem.

Temporal optimization and parallelization of platform-dependent operations is also being performed successfully at the task execution level. Keith et al. (2009) employ a temporal network that describes platform constraints to reorder and optimize the manipulator trajectories specified by a sequential plan. Konečnỳ et al. (2014) separate the strategic planning layer that only handles an abstract domain conceptualization from the detailed execution strategy that makes a plan executable on a real robot. However, the *Consistency Based Execution Monitoring* directly maps abstract, but fully grounded plan elements to a domain-specific execution strategy, without specifying an explicit platform model.

Kunze, Roehm, and Beetz (2011) introduce the Semantic Robot Description Language (SRDL) to bridge the gap between purely kinematic description languages and the more abstract level at which task specifications are usually formulated. They leverage the Web Ontology Language (Bechhofer et al. 2004) to model how domain-specific actions depend on platform-specific components that are required to realize them. Waibel et al. (2011) use SRDL to implement a shared knowledge base that allows robots to improve their search and execution strategies with previous observations possibly made by other robots. In this case, the knowledge base covers both platform-specific and domain-specific knowledge within a common deduction engine based on Description Logic (Baader 2003). The works based on SRDL are related to our work in their purpose, but differ significantly in that the SRDL model is purely a translation layer that sits between the abstract action plan and the executive layer. As such, SRDL specifications cannot be used to modify execution strategies at runtime, and thus cannot be used to dynamically deduce error recovery strategies. Mansouri and Pecora (2016) describe a constraint-based approach to hybrid reasoning with a meta-CSP that describes the different types of knowledge. The CSP is solved by a meta-solver that combines different kinds of reasoners. CHIMP (Stock et al. 2015) uses HTNs to solve such constraint-based hybrid reasoning tasks. HTN-based task decomposition approaches often model platform details as part of the planning problem. Dvorák et al. (2014) limit the problem size by delegating execution monitoring to a PRS subsystem with a simple success/failure interface.

Based on the Situation Calculus (McCarthy and Hayes 1969), the action language GOLOG allows a programmer to intermix imperative programming with planning on a logically formulated domain model (Levesque and Lakemeyer 2008). READYLOG (Ferrein and Lakemeyer 2008) extends the search functionality of GOLOG to allow for decision-theoretic planning. Finzi and Pirri (2005) provide a theoretical integration of the Situation Calculus with temporal constraints. De Giacomo, Reiter, and Soutchanski (1998) define an execution monitor in Golog that allows to react to unexpected changes during execution. Hofmann et al. (2016) interleave PDDL-based planning with Golog-based execution for monitoring purposes. Schiffer, Wortmann, and Lakemeyer (2010) describe an online transformation of a READYLOG program by inserting actions to satisfy qualitative temporal platform-specific constraints, under the assumption that agent domain and platform domain are disjunct.

Approach

Our goal is to design a framework that allows the user to formulate a platform constraint model that describes internal and external dependencies of component states, both in terms of hardware and software. An agent framework can then turn an abstract plan into a platform-specific execution and monitoring strategy that satisfies these constraints. This allows a separation of the high-level program from the specific platform properties while complying with the platform constraints. In the following, we present the different components of such a framework.

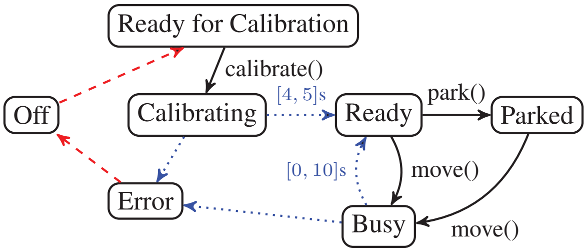


Figure 1: A finite state machine as a platform model for the Katana arm with three types of transitions: agent actions (black/solid), system events (blue/dotted), exogenous events (red/dashed). The edges are annotated with their action and expected time bounds.

Platform Models

Figure 1 shows an example for a model of a robotic manipulator arm, the Katana. Before the Katana arm can be used, it needs to be calibrated. Initially, the arm is turned off. It can only start its calibration process from a specific calibration position, so a human assistant must move the arm into the right position and then turn it on, which brings the arm into the state *Ready for Calibration*. From that state, the agent can decide to start the calibration. Note that this usually does not happen automatically, because the agent first has to make sure that it is in a location that allows an arm calibration, and second it may not need the arm at all. Since calibration is time-consuming, it should only be done if the arm is actually required. When the calibration is finished, the component driver triggers a transition to either the *Error* state or the *Ready* state. Similar to Schiffer, Wortmann, and Lakemeyer (2010), we model system components as state automata. But as the example in Fig. 1 shows, we need to differentiate between different kinds of transitions: 1. actions by the agent (black), 2. events triggered by the system (blue), 3. exogenous events (red).

Suitable Automata Models The platform model shown in Figure 1 is a finite state automaton with multiple edge types. However, more expressive automata models may be required to represent platform components. Consider a navigation stack that depends on the states of several low-level components, e.g., collision avoidance and localization. Each of these components will be modeled separately, but we might also want to formulate constraints on composite states covering multiple components. Hierarchical state machines as described in Girault, Lee, and Lee (1999) may be suitable to formulate such component-level abstractions. Timed transitions, such as the transition $\textit{Calibrating} \rightarrow \textit{Ready}$ may be modeled with timed automata (Alur and Dill 1994). While we will not change the foundation of our high-level reasoning, i.e., a situation calculus-based framework, we might consider a Petri-Net-based model such as the one described by Ziparo et al. (2011) for the component description as well.

Constraints

Platform constraints define properties that must always hold during the execution of the action plan. Based on Figure 1 and using Allen's interval relations (Allen 1983), we can define multiple constraints that must hold for the arm:

1. To calibrate the arm, the robot must be *at* a *free* location (i.e., a location without close objects).

$$free(at(x))\ \textbf{during}\ state(arm) = Calibrating$$

2. When starting to pick up an object, the arm must be ready or parked.

$$state(arm) = Ready\ \textbf{meets}\ pickup(x) \vee$$
$$state(arm) = Parked\ \textbf{meets}\ pickup(x)$$

3. Whenever the robot is moving, the arm must be parked.

$$state(arm) = Parked\ \textbf{during}$$
$$state(navigation) = Moving$$

Quantitative Temporal Constraints The examples above are qualitative temporal constraints. However, some components also require quantitative temporal constraints. Consider an RGBD camera that is used for perception. We can formulate the following constraints about the camera:

1. The camera needs some time to initialize, and therefore needs to be started one second before it can be used:

$$state(camera) = Running\ \textbf{before}_{\geq 1s}\ detect(x)$$

2. On the other hand, image processing is expensive, and thus should only be turned on if it is actually used within the next two seconds:

$$state(camera) \neq Running\ \textbf{unless}_{\leq 2s}\ detect(x)$$

The constraints above will be formulated in a temporal extension of the Situation Calculus and may refer to states of system components, fluents, and actions. While previous work only allowed qualititative temporal constraints (Schiffer, Wortmann, and Lakemeyer 2010), we want to allow for quantitative temporal constraints. In order to do so, we will extend the Situation Calculus based on Reiter (1996) and Gabaldon (2003) with qualitative and quantitative temporal aspects and embed the Metric Interval Temporal Logic (MITL) (Alur, Feder, and Henzinger 1996) into the Situation Calculus.

Events, Temporal Constraints, and Concurrency

The model of the Katana arm shown in Figure 1 has three kinds of edges: 1. Action edges that are directly triggered by the agent and are therefore under agent control, 2. Events that are triggered by the component itself, e.g. to end a durative action, 3. Exogenous events that are triggered by an external participant not under the agent's direct control, e.g., a human. Previously, both kinds of events were modeled as explicit exogenous actions with respective waiting actions. In our approach, we want to make use of concurrency in Golog with the *waitFor* construct (Grosskreutz and Lakemeyer 2003).

If we want to use the model of a system component to plan for a certain system configuration, e.g., a calibrated arm, we need to know about *expected* events. As an example, if the agent decides to start the calibration, it expects the calibration to finish successfully. If this was not the case, the agent could not cause state changes of system components in a meaningful way, as the outcome of any event transitions would be unknown. In addition to the information which transition is to be expected, we also annotate system events with expected time bounds. This allows the agent not only to reason about which event will occur, but also when it will occur. In the Katana example, we annotate the edge $Calibrating \rightarrow Ready$ with the expected time bounds $[4, 5]$, i.e., we expect the calibration to take at least four and at most five seconds. This way, the agent knows that it needs to start the calibration at least five seconds before it can use the arm.

Action Plan Transformation & Constraint Satisfaction

Given a platform-specific constraint model, an abstract action plan can be transformed into a platform-specific action plan that satisfies all constraints. To create such a plan, first the Golog interpreter determines an abstract action plan as usual. Next, the constraints are transformed into constraint networks (Dechter, Meiri, and Pearl 1991; Meiri 1996). In contrast to Finzi and Pirri (2005), we will not make use of *timelines*, but instead restrict our approach to interleaved and possibly true concurrency in order to allow a simpler formalization. Additionally, our approach will support quantitative constraints. The resulting constraint network will be evaluated with existing constraint solvers. A solution of the constraint network will determine the order of events with their interval limits. Platform constraints, e.g., $state(arm) = Ready$, must be transformed into actions by determining a suitable action sequence based on the platform model. The method of determining this action sequence depends on the underlying state machine model. For a simple state machine as shown in Figure 1, the actions can be determined by searching for a sequence of transitions that result in the desired state. For other, more expressive models, more complex methods may be necessary.

In some cases, such as the calibration of the Katana arm, inserting a single action may suffice. In other cases, the original action plan must be modified, e.g. to actively seek out localization features before some delicate manipulation task can be performed. Thus, a clear separation of the abstract agent and the plan transformation is not always possible and significant modifications of the original plan may be necessary. For this reason, the transformation of the abstract action plan into an executable plan will be part of the high-level agent and implemented within the Golog interpreter.

Conclusion

We presented a concept for an agent system with an explicit model of the robotic platform and its constraints. The robotic platform is modeled with state automata based on timed automata and hierarchical state machines and allows multiple transition types for agent actions, system events, and exogenous events. Based on these models, the user can formulate constraints in an extension of the Situation Calculus, which allows to define platform-specific, quantitative temporal constraints. During execution, the abstract action plan is modified to satisfy all constraints of the underlying platform. The proposed agent system allows the user to separate behavior control and platform management while taking into account that the constraints may require significant changes to the abstract action plan, which are handled by the agent system during execution.

Acknowledgments

This work was supported by the German National Science Foundation (DFG) under grant numbers GL-747/23-1 and FE 1077/4-1.

References

Allen, J. F. 1983. Maintaining knowledge about temporal intervals. *Commun ACM* 26(11):832–843.

Alur, R., and Dill, D. L. 1994. A theory of timed automata. *Theoretical Computer Science* 126(2):183–235.

Alur, R.; Feder, T.; and Henzinger, T. A. 1996. The benefits of relaxing punctuality. *J ACM* 43(1):116–146.

Baader, F. 2003. *The description logic handbook: Theory, implementation and applications*.

Bechhofer, S.; van Harmelen, F.; Hendler, J.; Horrocks, I.; McGuinness, D. L.; Patel-Schneider, P. F.; and Stein, L. A. 2004. OWL Web Ontology Language Reference. Technical report, W3C.

De Giacomo, G.; Reiter, R.; and Soutchanski, M. 1998. Execution Monitoring of High-Level Robot Programs. *Proc. of the 6th Int'l Conf. on Knowledge Representation and Reasoning (KR)*.

Dechter, R.; Meiri, I.; and Pearl, J. 1991. Temporal constraint networks. *Artificial Intelligence* 49(1-3):61–95.

Dvorák, F.; Barták, R.; Bit-Monnot, A.; Ingrand, F.; and Ghallab, M. 2014. Planning and acting with temporal and hierarchical decomposition models. In *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on*, 115–121. IEEE.

Ferrein, A., and Lakemeyer, G. 2008. Logic-based robot control in highly dynamic domains. *Robotics and Autonomous Systems* 56(11):980–991.

Finzi, A., and Pirri, F. 2005. Representing flexible temporal behaviors in the situation calculus. In Kaelbling, L. P., and Saffiotti, A., eds., *Proc. of the 19th Int'l Joint Conf. on Artificial Intelligence (IJCAI-05)*, 436–441.

Gabaldon, A. 2003. Compiling control knowledge into preconditions for planning in the situation calculus. In Gottlob, G., and Walsh, T., eds., *Proc. of the 18th Int'l Joint Conf. on Artificial Intelligence (IJCAI-03)*, 1061–1066.

Girault, A.; Lee, B.; and Lee, E. A. 1999. Hierarchical finite state machines with multiple concurrency models. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 18(6):742–760.

Grosskreutz, H., and Lakemeyer, G. 2003. ccGolog – a logical language dealing with continuous change. *Logic Journal of the IGPL* 11(2):179–221.

Hofmann, T.; Niemueller, T.; Claßen, J.; and Lakemeyer, G. 2016. Continual Planning in Golog. In *Proc. of the 30th Conf. on Artificial Intelligence (AAAI)*.

Jónsson, A. K.; Morris, P. H.; Muscettola, N.; Rajan, K.; and Smith, B. D. 2000. Planning in interplanetary space: Theory and practice. In Czhien, S.; Kambhampati, S.; and Knoblock, C. A., eds., *Proc. of the 15th Int'l Conf. on Artificial Intelligence Planning Systems, (AIPS-00)*, 177–186.

Keith, F.; Mansard, N.; Miossec, S.; and Kheddar, A. 2009. From discrete mission schedule to continuous implicit trajectory using optimal time warping. In *Proc. of the 19th Int'l Conf. on Automated Planning and Scheduling*.

Kim, P.; Williams, B. C.; and Abramson, M. 2001. Executing reactive, model-based programs through graph-based temporal planning. In *Proc. of the 17th Int'l Joint Conf. on Artificial Intelligence (IJCAI-01)*, 487–493.

Konečnỳ, Š.; Stock, S.; Pecora, F.; and Saffiotti, A. 2014. Planning domain+ execution semantics: A way towards robust execution? In *2014 AAAI Spring Symposium Series – Qualitative Representations for Robots*.

Kunze, L.; Roehm, T.; and Beetz, M. 2011. Towards semantic robot description languages. In *IEEE Int'l Conf. on Robotics and Automation (ICRA 2011)*, 5589–5595.

Lemai, S., and Ingrand, F. 2004. Interleaving temporal planning and execution in robotics domains. In McGuinness, D., and Ferguson, G., eds., *Proc. of the 19th Nat'l Conf. on Artificial Intelligence (AAAI-04) and 16th Conf. on Innovative Applications of Artificial Intelligence (IAAI-04)*, 617–622.

Levesque, H., and Lakemeyer, G. 2008. Chapter 23 Cognitive Robotics. In Frank van Harmelen, V. L., and Porter, B., eds., *Handbook of Knowledge Representation*, volume 3 of *Foundations of Artificial Intelligence*. 869–886.

Mansouri, M., and Pecora, F. 2016. A robot sets a table: a case for hybrid reasoning with different types of knowledge. *Journal of Experimental & Theoretical Artificial Intelligence* 28(5):801–821.

McCarthy, J., and Hayes, P. 1969. Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B., and Michie, D., eds., *Machine Intelligence 4*. 463–502.

Meiri, I. 1996. Combining qualitative and quantitative constraints in temporal reasoning. *Artificial Intelligence* 87(1-2):343–385.

Reiter, R. 1996. Natural actions, concurrency and continuous time in the situation calculus. In Aiello, L. C.; Doyle, J.; and Shapiro, S. C., eds., *Proc. of the 5th Int'l Conf. in Principles of Knowledge Representation and Reasoning (KR-96)*, 2–13.

Schiffer, S.; Wortmann, A.; and Lakemeyer, G. 2010. Self-Maintenance for Autonomous Robots controlled by ReadyLog. In Ingrand, F., and Guiochet, J., eds., *Proc. of the 7th*

IARP Workshop on Technical Challenges for Dependable Robots in Human Environments (DRHE2010), 101–107.

Stock, S.; Mansouri, M.; Pecora, F.; and Hertzberg, J. 2015. Online task merging with a hierarchical hybrid task planner for mobile service robots. In *Proc. of the Int'l Conf. on Intelligent Robots and Systems (IROS)*, 6459–6464.

Tsamardinos, I.; Muscettola, N.; and Morris, P. H. 1998. Fast transformation of temporal plans for efficient execution. In *Proc. of the 15th Nat'l Conf. on Artificial Intelligence (AAAI-98) and 10th Innovative Applications of Artificial Intelligence Conf. (IAAI-98)*, 254–261.

Waibel, M.; Beetz, M.; Civera, J.; D'Andrea, R.; Elfring, J.; Galvez-Lopez, D.; Haussermann, K.; Janssen, R.; Montiel, J.; Perzylo, A.; Schiessle, B.; Tenorth, M.; Zweigle, O.; and van de Molengraft, R. 2011. RoboEarth - A World Wide Web for Robots. *IEEE Robotics Automation Magazine* 18(2):69–82.

Wisspeintner, T.; Van Der Zant, T.; Iocchi, L.; and Schiffer, S. 2009. Robocup@ home: Scientific competition and benchmarking for domestic service robots. *Interaction Studies* 10(3):392–426.

Ziparo, V. A.; Iocchi, L.; Lima, P. U.; Nardi, D.; and Palamara, P. F. 2011. Petri net plans. *Autonomous Agents and Multi-Agent Systems* 23(3):344–383.

Learning to Act in Partially Structured Dynamic Environment

Chen Huang,[1] Lantao Liu,[2] Gaurav Sukhatme[1]
[1]Department of Computer Science at the University of Southern California, Los Angeles, CA 90089, USA
E-mail: {huan574, gaurav}@usc.edu
[2]Intelligent Systems Engineering Department at Indiana University - Bloomington
Bloomington, IN 47408, USA. E-mail: lantao@iu.edu

Abstract

We investigate the scenario that a robot needs to reach a designated goal after taking a sequence of appropriate actions in a non-static environment that is partially structured. One application example is to control a marine vehicle to move in the ocean. The ocean environment is dynamic and the ocean waves typically result in strong disturbances that can disturb the vehicle's motion.

Modeling such dynamic environment is non-trivial, and integrating such model in the robotic motion control is particularly difficult. Fortunately, the ocean currents usually form some local patterns (e.g. vortex) and thus the environment is partially structured. The historically observed data can be used to train the robot to learn to interact with the ocean flow disturbances. In this paper we propose a method that applies the deep reinforcement learning framework to learn such partially structured complex disturbances. Our preliminary results show that, by training the robot under artificial and real ocean disturbances, the robot is able to successfully act in complex and spatiotemporal environments.

Introduction and Related Work

Acting in unstructured environments can be challenging especially when the environment is dynamic and involves continuous states. We study the goal-directed action decision-making problem where a robot's action can be disturbed by environmental disturbances such as the ocean waves or air turbulence.

To be more concrete, consider a scenario where an underwater vehicle navigates across an area of ocean over a period of a few weeks to reach a goal location. Underwater vehicles such as autonomous gliders currently in use can travel long distances but move at speeds comparable to or slower than, typical ocean currents [Wynn et al., Smith et al.]. Moreover, the disturbances caused by ocean eddies oftentimes are complex to be modeled. This is because when we navigate the underwater (or generically aquatic) vehicles, we usually consider long term and long distance missions, and during this process the ocean currents can change significantly, causing spatially and temporally varying disturbances. The ocean currents are not only complex in patterns, but are also strong in tidal forces and can easily perturb the underwater vehicle' motion, causing significantly uncertain action outcomes.

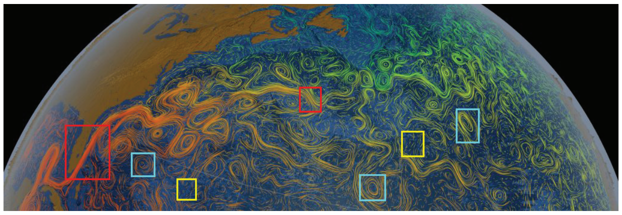


Figure 1: Ocean currents consist of local patterns (source: NASA). Red box: uniform pattern. Blue box: vortex. Yellow box: meandering

In general, such non-static and diverse disturbances are a reflection of the unstructured natural environment, and oftentimes it is very difficult to accurately formulate the complex disturbance dynamics using mathematical models. Fortunately, many disturbances caused by nature are seasonal and can be observed, and the observation data is available for some time horizons. For example, we can get the forecast, nowcast, and hindcast of the weather including the wind (air turbulence) information. Similarly, the ocean currents information can also be obtained, and using such data allows us to train the robot to learn to interact with the ocean currents.

Recently, studies on deep and reinforcement learning have revealed a great potential for addressing complex decision problems such as game playing [Mnih et al., Silver et al., Oroojlooyjadid et al.].

We found that there are certain similarities between our marine robots decision-making and the game playing scenarios if one regards the agent's interacting platform/environment here is the nature instead of a game. However, one general critical challenge that prevents robots from using deep learning is the lack of sufficient training data. Indeed, using robots to collect training data can be extremely costly (e.g., in order to get one set of marine data using on-board sensors, it is not uncommon that a marine vehicle needs to take a few days and traverse hundreds of miles). Also, modeling a vast area of environment can be computationally expensive.

Fortunately, oftentimes the complex-patterned disturbance can be characterized by local patches, where a single patch may possess a particular disturbance pattern (e.g., a vortex/ring pattern) [Oey, Ezer, and Lee], and the total number of the basic patterns are enumerable. Therefore, we are motivated by training the vehicle to learn those local patches/patterns offline so that during the real-time mission, if the disturbance is a mixture of a subset of those learned patterns, the vehicle can take advantage of what it has learned to cope with it easily, thus reducing the computation time for online action prediction and control. We use the iterative linear quadratic regulator [Li and Todorov] to model the vehicle dynamics and control, and use the policy gradient framework [Levine and Koltun] to train the network. We tested our method on simulations with both artificially created dynamic disturbances as well as from a history of ocean current data, and our preliminary results show that the trained robot achieved satisfying performance.

Technical Approach

We use the deep reinforcement learning framework to model our decision-making problem. Specifically, we use s, a to denote the robot's state and action, respectively. The input of the deep network is the disturbance information which is typically a vector field. Our goal is to obtain a stochastic form of policy $\pi_\theta(s,a) = \mathrm{P}(a|s,\theta)$ paramterized by θ (i.e., weights of the neural network) that maximizes the discounted, cumulative reward $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}$, where T is a horizon term specifying the maximum time steps and r_t is the reward at time t and γ is a discounting constant between 0 and 1 that ensures the sum converges. A deep convolutional neural network is used to approximate the optimal action-value function $Q^*(s,a) = \max_\pi \mathrm{E}[R_t|s_t, a_t, \pi]$. More details of the basic model can be found in [Mnih et al.].

Network Design

Since the ocean currents data over a period is available, we build our neural network with an input that integrates both the ocean (environmental) and the vehicle's states. The environmental state here is a vector field representing the ocean currents (their strengths and directions). Fig. 2 shows the structure of the neural network.

Specifically, the input consists of two components: environment and vehicle states. The environment component has three channels, where the first two channels convey information of the x-axis and y-axis of the disturbance vector field. Since in the environment we need to define goal states, and there may be obstacles, thus, we use a third channel to capture such information. In greater detail, we assume that each grid of the input map has three forms: it can be occupied by obstacle (we set its value -1), or be free/empty for robot to transit to (with value 0), or be occupied by the robot (with value 1). The other component of the input is a vector that contains vehicle state information, including the vehicle's velocity and its direction towards the goal. Note that we do not include the robot's position in input because we want the robot to be sensitive only to environmental dynamics but not to specific (static) locations.

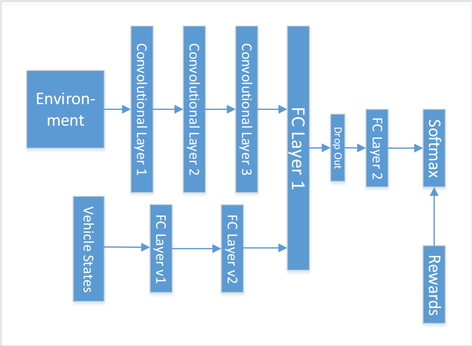


Figure 2: Neural Network Structure

The design of internal hidden layers is depicted in Fig. 2. The front 3 convolutional layers process the environment information, while the vehicle states begin to be combined starting from the first fully connected (FC) layer. The reason of such a design lies in that, the whole net could be regarded as two sub-nets that are not strongly correlated: one subnet is used to characterize features of disturbances, which is analogous to that of image classification; the other sub-net is a decision component for choosing the best action strategy. In addition, such separation of input can reduce the number of parameters so that the training process can be accelerated.

After each convolutional layer a max-pool is applied. The vehicle states will pass through 2 FC layers, and then are combined with the environmental component output from convolutional layer 3 as the input to a successive FC Layer 1. Between FC Layer 1 and 2 there exists a drop-out layer to avoid overfitting. The Softmax layer is used to normalize outputs for generating a probability distribution that can be used for sampling future actions. Additionally, the *loss funciton* is calculated using this probability distribution as well as the actual rewards.

Loss Function and Reward

We employ the policy gradient framework for solution convergence. With the stochastic policy $\pi_\theta(s,a)$ and the Q-value $Q_{\pi_\theta}(s,a)$ for the state-action pair, the policy gradient of loss function is $L(\theta)$ can be defined as follows:

$$\nabla_\theta L(\theta) = \mathbb{E}_{\pi_\theta}\left[Q_{\pi_\theta}(s,a)\nabla_\theta \log \pi_\theta(s,a)\right]. \quad (1)$$

To improve the sampling efficiency and accelerate the convergence, we adopt the *importance sampling* strategy using guided samples [Levine and Koltun].

With the objective of reaching the designated goal, our rewarding mechanism is therefore to minimize the cost from start to goal. The main idea is to reinforce with a large positive value for those correct actions that lead to reaching the goal quickly, and punish those undesired actions (e.g., those take long time or even fail to reach the goal) with small or even negative values. Formally, we define the reward r of each trial/episode as:

$$r = \begin{cases} r_s, & \text{succeeded,} \\ -(\alpha r_s + (1-\alpha) r_d), & \text{failed.} \end{cases} \quad (2)$$

where

$$r_s = \frac{1}{\sum_t \pi_\theta(s,a)||p_t - p_G||_2}, \qquad (3)$$

$$r_d = 1 - e^{-D_{min}}. \qquad (4)$$

where $||p_t - p_G||_2$ denotes the distance from the t-th step position to the goal state, and $D_{min} = \min_t ||p_t - p_G||_2$ is the minimum such distance along the whole path. The term r_s in Eq. (3) evaluates the state with respect to the goal state, whereas the term r_d in Eq. (4) summarizes an evaluation over the entire path. Coefficient $\alpha \in [0,1]$ is an empirical value to scale between r_s and r_d so that they contribute about the same to the total reward r. In our experiments α is set to 0.9.

Offline Training and Online Decision-Making

We train the robot by setting different starting and goal positions in the disturbance field, and the *experience replay* [Mnih et al., Riedmiller] mechanism is employed. Specifically, we define an *experience* as a 3-tuple (s, a, r) consisting of state s, action a, and reward r. The idea is to store those experiences obtained in the past into a dataset. Then during the reinforcement learning update process, a mini-batch of experiences is sampled from the dataset each time for training. The process of training is described in Algorithm 1, which can be summarized into four steps.

1. Following incumbent action policies, sample actions and finish a trial path or an episode.
2. Upon completion of each episode, obtain corresponding rewards (a list) according to whether the goal is reached, and assign the rewards to actions taken on that path.
3. Add all these experiences into dataset. If the dataset has exceeded the maximum limit, erase as many as the oldest ones to satisfy the capacity.
4. Sample a mini-batch of experiences from the dataset. This batch should include the most recent path. Then shuffle this batch of data and feed them into the neural network for training. If current round number is less than the max training rounds, go back to step 1.

With the offline trained results, the decision-making is straightforward: only one forward propagation of the network with small computational effort is needed. This also allows us to handle continuous motion and unknown states.

Results

We validated the method in the scenario of marine robot goal-driven decision-making, where the ocean disturbances vary both spatially and temporally.The simulation environment was constructed as a two dimensional ocean surface, and the spatiotemporal ocean currents are external disturbances for the robot and are represented as a vector field, with each vector representing the water flow speed captured at a specific moment in a specific location.

The robot used in simulation is a underwater glider with a kinematic motion model with state $z = (x, y, \phi)$ including the vehicle's position and orientation in the world frame, respectively. Since the behavior of the vehicle on the 2D ocean surface is similar to that of the ground mobile robot, thus we opt to use a Dubins car model to simulate its motion. (Similar settings can be found in [Yao, Wang, and Su, Mahmoudian and Woolsey].) The dynamics can be written as:

$$\dot{x} = v\cos\phi, \quad \dot{y} = v\sin\phi, \quad \dot{\phi} = \omega, \qquad (5)$$

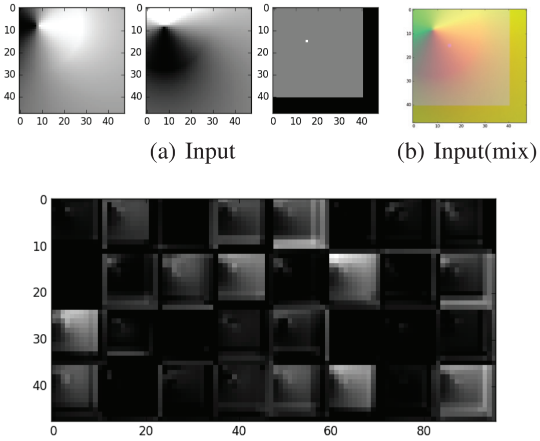
where control inputs $u = (v, \omega)$ are the vehicle's net speed and turning rate, respectively. The dynamics are obvious nonlinear and in the discrete time case are denoted as $z_{t+1} = f(z_t, u_t)$. Such non-linear control problem can be solved using the iterative Linear Quadratic Regulator (iLQR) [Li and Todorov].

Network Training

We use Tensorflow [Abadi et al.] to build and train the network described in Fig. 2. In our experiments, the input vec-

Algorithm 1: Training

```
$round \leftarrow 0$
while $round < n$ do
    Obtain reward $List\langle s,a\rangle$ of each episode.
    $experiences \leftarrow \varnothing$
    for all $\langle s,a\rangle \in List\langle s,a\rangle$ do
        $r \leftarrow get\_reward(s,a)$
        $experiences \leftarrow experiences \bigcup \langle s,a,r\rangle$
    end for
    $subset \leftarrow experiences$
    pad up $subset$ to batch size with data from dataset
    store $experiences$ into dataset
    shuffle $subset$
    feed $subset$ into neural network
    perform back propagation
    $round \leftarrow round + 1$
end while
```



(a) Input (b) Input(mix)

(c) Convolutional Layer 3

Figure 3: Illustration of disturbance features captured by hidden layer

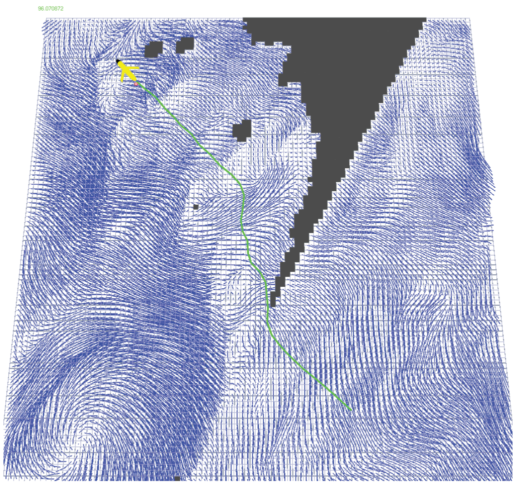


Figure 4: Demonstration of the ocean currents and a path of the robot

tor field map is 48×48, and the size of dataset for action replay is set to 10000. The learning rate is $1e - 6$, and the batch size we used for each iteration is 500. We also set the length of each episode as 1000 steps.

Fig. 3 shows some features extracted from internal layers of the network. Fig. 3(a) illustrates the feature of a random disturbance vector field. Specifically, the first two channels of Fig. 3(a) are x and y components of the vector field, and the grey-scale color represents the strength of disturbance. The third channel of Fig. 3(a) is a pixel map that contains the goal point (white dot) and obstacle information (black borders).

Other grey grids denote free place. Fig. 3(b) shows a mixed view of the features, with three channels colored in red, green and blue, respectively. The picture depicts a local vortex pattern with the vortex center located near the upper left corner. Fig. 3(c) shows outputs of convolutional layer 3, from which we can observe that the hidden layers extract some local features.

Evaluations

We implemented two methods: one belongs to the control paradigm and we use the basic iLQR to compute the control inputs; the other one is the deep reinforcement learning (DRL) framework that employs the guided policy mechanism, where the policy is guided by (and combined with) the iLQR solving process [Levine and Koltun].

Artificial Disturbances We first investigate the method using artificially generated disturbances. We tested different vector fields including vortex, meandering, uniform, and centripetal patterns.

For different trials, we specify the robot with different start and goal locations, and the *goal reaching rate* is calculated by the times of success divided by total number of simulations.

The results in Table 1 show that within given time limits, both the iLQR and DRL methods lead to a good success rate, and particularly the DRL performs better in complex environments like the vortex field; whereas the iLQR framework has a slightly better performance in relatively mild environments where current speed is low, like the meander disturbance field.

Then, we test the average time costs, as shown in Table 2. The results reveal that the trials using iLQR tend to use less time than those of the DRL method. This can be due to the "idealized" artificial disturbances with simple and accurate patterns, which can be precisely handled by the traditional control methodology.

Disturbance pattern	Method	Num of trials	Num of success	Success rate
Vortex	DRL	50	48	0.96
	iLQR	50	46	0.92
Meander	DRL	50	49	0.98
	iLQR	50	50	1.00
Uniform	DRL	50	49	0.98
	iLQR	50	48	0.96
Centripetal	DRL	50	49	0.98
	iLQR	50	48	0.96

Table 1: Simulation with artificially generated disturbances

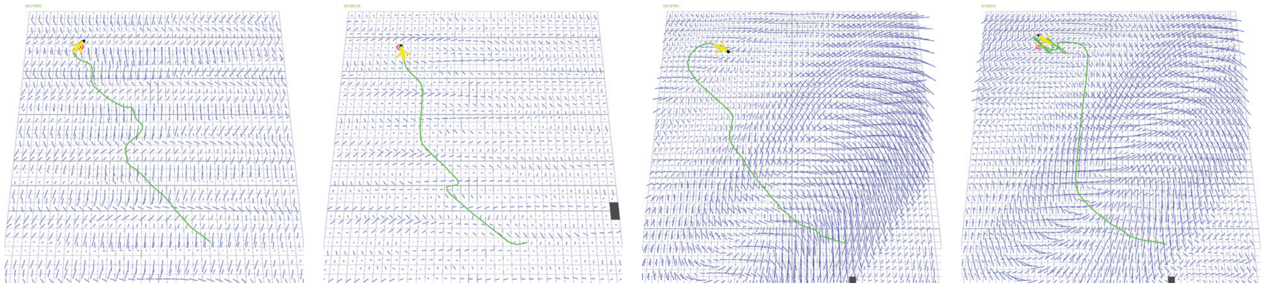
Ocean Data Disturbances In this part of evaluation, we use ocean current data obtained from the California Regional Ocean Modeling System (ROMS) [Shchepetkin and McWilliams]. The ocean data along the coast near Los Angeles is released every 6 hours and a window of 30 days of data is maintained and retrievable [Chao].

An example of ocean current surface can be visualized in Fig. 4, which also demonstrates a robot's path from executing our training result.

Because the raw ROMS ocean data covers a vast area and practically it requires several days for the robot to travel through the whole space, thus, we randomly cropped local areas to evaluate our training results. Fig. 5 demonstrates a few paths generated in such randomly selected areas.

Similar to the evaluation process for the artificial disturbances, we also looked into those aforementioned performances under the real ocean disturbances. We then evaluate the success rate and time cost, and Table. 3 shows the results (robot speed does not scale to map). Fig. 5 gives a more friendly visualization of those three areas used in our experiments. The results indicate that in most cases the DLR performs better than the basic iLQR strategy.

Fig. 5(c) and 5(d) show scenarios that can be challenging due to strong vortexes. Fig. 5(c) shows that by selecting a good path going around the vortex, the robot successfully reached the goal state. Note, in the area 3 of Fig. 5(d), a very curvy path (e.g., near the goal point) could occur due to some strong vortex in certain local areas. In this example, the ocean current around the goal area has a speed approximately equal to (or even greater than) the robot's maximal speed, but is against the robot's moving direction, so that the robot cannot easily proceed, and both DRL and iLQR eventually failed to reach the goal in this situation. A possible



(a) area1 (b) area2 (c) area3 (d) area3

Figure 5: Examples of robot paths under different spatiotemporal disturbance patterns.

Pattern	Method	Num of trials	Average time cost
Vortex	DRL	50	20.549
	iLQR	50	14.811
Meander	DRL	50	16.926
	iLQR	50	15.367
Uniform	DRL	50	17.667
	iLQR	50	17.803
Centripetal	DRL	50	20.220
	iLQR	50	14.792

Table 2: Average time cost under artificial disturbances

Area	Method	Num of trials	Success rate	Average time cost
Area 1	DRL	15	1.00	13.787
	iLQR	15	0.93	16.375
Area 2	DRL	15	1.00	14.998
	iLQR	15	1.00	15.530
Area 3	DRL	15	0.60	22.875
	iLQR	15	0.80	19.546

Table 3: Average time cost under ocean disturbances

solution is to manipulate the robot's maximal speed to be larger (this however may be against the reality).

From Table 1 to Table 3, we can conclude that the DRL framework is particularly capable of handling complex and (partially) unstructured environments.

Conclusions

In this paper we investigate applying the deep reinforcement learning framework for robotic learning and acting in partially-structured environments. We use the scenario of marine vehicle decision-making under spatiotemporal disturbances to demonstrate and validate the framework. We show that the deep network well characterizes local features of varying disturbances. By training the robot under artificial and real ocean disturbances, our simulation results indicate that the robot is able to successfully and efficiently act in complex and partially structured environments.

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; and Zheng, X. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Chao, Y. 2017. Regional ocean model system. http://www.sccoos.org/data/roms-3km/.

Levine, S., and Koltun, V. 2013. Guided policy search. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, 1–9.

Li, W., and Todorov, E. 2004. Iterative linear quadratic regulator design for nonlinear biological movement systems. In *ICINCO (1)*, 222–229.

Mahmoudian, N., and Woolsey, C. 2008. Underwater glider motion control. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, 552–557. IEEE.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Oey, L.-Y.; Ezer, T.; and Lee, H.-C. 2005. Loop current, rings and related circulation in the gulf of mexico: A review of numerical models and future challenges. *Circulation in the Gulf of Mexico: Observations and models* 31–56.

Oroojlooyjadid, A.; Nazari, M.; Snyder, L. V.; and Takác, M. 2017. A deep q-network for the beer game with partial information. *CoRR* abs/1708.05924.

Riedmiller, M. 2005. Neural fitted q iteration-first experiences with a data efficient neural reinforcement learning method. In *ECML*, volume 3720, 317–328. Springer.

Shchepetkin, A. F., and McWilliams, J. C. 2005. The regional oceanic modeling system (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling* 9(4):347–404.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; Chen, Y.; Lillicrap, T.; Hui, F.; Sifre, L.; van den Driessche, G.; Graepel, T.; and Hassabis, D. 2017. Mastering the game of go without human knowledge. *Nature* 550(7676):354–359.

Smith, R. N.; Schwager, M.; Smith, S. L.; Jones, B. H.; Rus, D.; and Sukhatme, G. S. 2011. Persistent ocean monitoring with underwater gliders: Adapting sampling resolution. *Journal of Field Robotics* 28(5):714 – 741.

Wynn, R. B.; Huvenne, V. A.; Bas, T. P. L.; Murton, B. J.; Connelly, D. P.; Bett, B. J.; Ruhl, H. A.; Morris, K. J.; Peakall, J.; Parsons, D. R.; Sumner, E. J.; Darby, S. E.; Dorrell, R. M.; and Hunt, J. E. 2014. Autonomous underwater vehicles (auvs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology* 352:451 – 468.

Yao, P.; Wang, H.; and Su, Z. 2015. Uav feasible path planning based on disturbed fluid and trajectory propagation. *Chinese Journal of Aeronautics* 28(4):1163–1177.

Represention, Use, and Acquisition of Affordances in Cognitive Systems

Pat Langley
Institute for the Study of Learning and Expertise
2164 Staunton Court, Palo Alto, CA 94306 USA

Mohan Sridharan,Ben Meadows
Department of Electrical & Computer Engineering
University of Auckland, Private Bag 92019
Auckland 1142 New Zealand

Abstract

We review the psychological notion of *affordances* and examine it anew from a cognitive systems perspective. We distinguish between environmental affordances and their internal representation, choosing to focus on the latter. We consider issues that arise in representing mental affordances, using them to understand and generate plans, and learning them from experience. In each case, we present theoretical claims that, together, form an incipient theory of affordance in cognitive systems. We close by noting related research and proposing directions for future work in this arena.

1 Introduction and Background

Intelligent agents, both human and artificial, often operate in the context of an external environment and interact with entities therein. The agent can interact effectively with these objects in some ways but not others. For instance, depending on its manipulators, an agent will be able to grasp, lift, or throw some items but not different ones. Similarly, it can sit or recline on some objects but not others. Gibson (1977) referred to such relationships as *affordances*, a term that has been widely adopted in perceptual psychology, human-computer interaction, and, more recently, AI and robotics.

Gibson viewed affordances as existing in the environment, but others have used the term, rather differently, to refer to internalized models of these relations. For example, Vera and Simon (1993) have proposed that they are encoded as symbol structures which the agent can use to guide its decision making. They mapped affordances onto both the condition sides of production rules and onto perceptual chunks to which they refer. More recently, Sahin et al. (2007) and Zech et al. (2017) have reviewed different formalizations in robotics, focusing on relations between agents and the environment. We will incorporate ideas from each of these earlier efforts in our own analysis.

In this paper we present a high-level theory of affordances that makes commitments about a number of key issues. Like Vera and Simon, we focus on internal representations of affordances that describe an agent's ability for action. However, we move beyond their treatment to make more specific statements about the role of affordances in intelligence, focusing in turn on issues of representation, performance, and learning. We propose theoretical postulates about affordances that we feel are promising, but we do not report implemented agents that incorporate these tenets or experimental evaluations of them, which we reserve for future work.

2 Representing Knowledge of Affordances

Because representation constrains both performance and learning, we should address first how an intelligent agent can encode affordances in memory and how they relate to other cognitive structures. We distinguish between grounded short-term elements, say a belief that the agent can lift a particular box, and generic long-term ones, say a predicate and associated rule that specifies the class of situations in which lifting is possible. The typical usage of 'affordance' focuses on the grounded version, but we maintain that such elements are always instances of generic structures, so the primary representational challenges concern encoding the latter.

We hypothesize two distinct forms of knowledge: *concepts* that denote classes of objects or relations among them; and *skills* that specify the conditions in which multi-step activities produce specific outcomes.[1] Skills refer to concepts when describing their conditions and effects, making the latter structures more basic than the former. This leads naturally to our first theoretical postulate:

- *Affordances are concepts that describe the class of situations and the characteristics of agents for which particular activities produce specific effects.*

In other words, they are reified predicates that link the structures of objects and the features of agents that can use those objects to achieve given ends. Affordances take the same form as other concepts, in that they specify a predicate with associated arguments and a set of conditions that describe when they hold. The key difference is that each affordance concept serves as the sole condition on a skill, indicating when the latter produces its associated effects. Conceptual memory also contains other concepts, such as ones that describe situations which result from a skill's application.

Note that we view affordances as three-way relationships among the way an object is used, structural aspects of that

[1] We have borrowed this disctintion from Li, Stacuzzi, and Langley's (2012) ICARUS architecture, but it has roots in psychology.

object, and characteristics of the agent that uses it. A typical hammer has a handle with a head on one end, but it cannot be used to drive a nail or spike unless the agent is strong enough to lift and swing it. This means that a sledge hammer may afford the hammering activity for some agents but not others. Some conditions in an affordance concept will be qualitative, but others will specify numeric relations, such as whether a tool's weight is less than what the agent can lift.

We also postulate that many affordances are matters of degree. Some handles are easier for a given agent to grasp than others, while some ladders are easier for that agent to climb. This suggests that logical definitions of concepts, often assumed in AI, are insufficient. Instead, we propose that:

- *Affordances are graded concepts that match situations to greater or lesser degrees*.

For instance, a hammer may be more or less usable by a person depending on the difference between its weight and what he can lift, among other factors. Probabilistic categories are one way to support graded behavior, but any approach that measures distance from a prototype or central tendency will suffice. Most work in this tradition has assumed attribute-value notations, but one can also define relational concepts that match to different degrees (e.g., Choi 2010).

Finally, treating affordances as reified conceptual predicates suggests another representational characteristic that, we hypothesize, is especially important for describing extended activities that involve multiple steps:

- *Complex affordances are decomposable into elements that denote different aspects of usability*.

For example, a tool has a hammering affordance when an agent can grasp its handle, lift it upward, and propel its flat head against the target. We can view each of these elements as a distinct 'subaffordance' that must hold, for a given agent and to a reasonable degree, to let the agent use a tool for its intended function. A hammer may be light enough for a person to lift, but it will not drive home a nail if its handle is so slippery that it flies out of his grasp or if its head is so narrow that it misses the target.

3 Using Knowledge of Affordances

Humans and other intelligent agents engage in two broad classes of knowledge-based cognition. One involves interpreting situations and events in the environment, in some cases the activities of other agents. For instance, we may observe someone stacking some boxes but appear to have difficulty lifting one that is too heavy. The simplest variant is intention recognition, which assigns an agent's behavior to some known category, such as picking up a hammer or stacking a box. A more complex version, plan understanding (e.g., Meadows et al. 2014), infers an agent's multi-step plan, including goals it aims to achieve. Our next claim involves two facets of this performance task:

- *Affordances enable both proposal of hypotheses during plan understanding and their evaluation*.

To clarify hypothesis creation, suppose that we observe someone holding a nail and reaching in the direction of two objects, a hatchet and a screwdriver. The hatchet's structure, specifically its handle and the flat side of its head, can be used to hammer the nail, suggesting this as a candidate intention. The latter occurs because the hatchet's description, obtained through perception and inference, matches the affordance conditions associated with hammering a nail. The screwdriver does not lend itself structurally to this activity, so it would not produce a comparable hypothesis.

The graded nature of affordances helps during evaluation of candidate explanations. Given a set of observations, some intentions and plans will be more plausible than others. For example, suppose we observe someone in a room picking up a shoe that has a flat heel. We might hypothesize that he plans to put the object on his foot or that he plans to use it to hammer a nail. The shoe can be used for both activities, but it matches the affordance concept for placing on a foot much better than it does the one for hammering. We can use this degree of match in our evaluation of the two hypotheses and conclude that the first alternative is more plausible.

The second performance task concerns generating activities that support one's goals. As before, the simplest cases involve selection of primitive actions, such as grasping a glass or lifting a held box. More complicated variants involve chaining sequences of actions into multi-step plans to achieve the agent's goals. This suggests another tenet:

- *Affordances aid both the proposal of actions during plan generation and their evaluation*.

For instance, suppose we want a nail embedded in a wall and we have two tools, a hatchet and a screwdriver. We might use means-ends analysis to propose a hammering activity that achieves the goal and then realize the hatchet, held in a particular orientation, satisfies the affordance concept for hammering, but the screwdriver does not. Or we might use forward chaining to identify which affordances match the current situation, retrieve their associated activities, and consider the resulting states. Hammering the nail with the reversed hatchet is an applicable action that achieves the goal, but no screwdriver-related activities are applicable. If the nail were a screw, the situation would be inverted.

Affordances can also influence evaluation of candidate intentions during the planning process. Suppose, again, that we want a nail embedded in the wall, and that we have generated two possible intentions: hammering the nail with a reversed hatchet and hammering it with a shoe. Both satisfy the relational conditions of the graded affordance for hammering, but the hatchet would match its specification better than the shoe. The reasons involve both the relative abilities for grasping the two tools and their capacities for driving the nail into the wall even when they are held firmly.

4 Acquiring Knowledge of Affordances

Now that we have discussed the representation and use of internal affordances, we can turn briefly to their acquistion from experience. Recall that affordance concepts describe the conditions under which an activity has a particular effect for an agent. The AI community has pursued two different approaches to learning about agents' activities that suggest a final theoretical postulate:

- *Primitive affordances are learned inductively whereas complex affordances are learned analytically.*

When an agent first interacts with a new object or situation, it has little knowledge on which to build. In response, learning the conditions under which an action will have desired effects – the affordance concept – is primarily empirical. For example, this can occur by attempting to grasp different objects, with induction comparing configurations of successful and unsuccesful cases (e.g., Shen and Simon 1989).

In contrast, acquisition of complex affordances occurs in the presence of existing components, enabling use of analytic methods like those used to determine conditions on macro-operators (Iba 1989). This involves composing the conditions of actions not satisfied by the effects of those that occur before them. For instance, if we have affordance concepts for grasping a hammer's handle, lifting it, and hitting a nail with its head, then each of these would appear as components of a complex affordance for hammering a nail. Interactions among these elements may require inductive refinement, but creation of an initial concept can occur analytically based on a single training case. Li et al. (2012) have adapted this compositional method to acquire definitions for new conceptual predicates, in some cases recursive ones, that serve as conditions on learned hierarchical skillls.

5 Related Research

Recent years have seen growing interest in internalized affordances within the AI and robotics communities. Horton, Chakraborty, and St. Amant (2012) review many of these efforts, which often use visual processing to classify objects as appropriate for actions. Sahin et al. (2007) and Zech et al. (2017) also offer insightful surveys of computational research on the topic. We should examine how our theoretical claims relate to the growing body of work in this area.

- *Affordances are concepts that map relations between situations and agents on the effects of actions.*

A review of the literature reveals that some aspects of this statement are widely accepted but not others. Treatments of affordances have always involved mapping objects or situations onto action relevance, and many efforts to learn such mappings produce conceptual descriptions or classifiers. However, the notion that affordances involve *interactions* between features of agents and features of objects has been much less common. Stoffregen (2003) provides an early and clear statement of this claim, but his treatment was informal and, to our knowledge, AI and robotics papers have only rarely incorporated his insight. We maintain that this important idea deserves more attention in the computational literature than it has received.

- *Affordances are graded concepts that match situations to greater or lesser degrees.*

Prior researchers have not discussed this idea directly. For instance, Sarathy and Scheutz (2016) describe an approach that uses probabilistic rules to infer affordances of objects for actions. Their framework shares our assumption that affordances are reified concepts, but not that these mental structures are graded. Zech et al. (2017) consider dynamic affordances that vary with changing properties of objects, but they remain Boolean in each case. They also suggest that agents choose among objects based on appropriateness to a given outcome, but stop short of proposing degrees of affordance. Of course, probabilistic approaches can predict how features of the agent and situation affect an action's chance of success, but graded affordances can also encode the time, effort, and difficulty of achieving an objective. Thus, this claim seems like an important contribution to the literature.

- *Complex affordances are decomposable into elements that denote different aspects of usability.*

This idea appears in a few places but has not been explored in detail. Zech et al. review a few papers that discuss a hierarchy of affordances, including Ellis and Tucker's (2000) experimental studies of 'micro-affordances' as 'potentiated components' of higher-level activities (e.g., turning a wrist while reaching for an object). However, computational researchers have generally focused on a single level of analysis. Therefore, the decomposition of complex affordances into simpler elements, and the compositional semantics it requires, is a notion that merits substantially more effort than the community has given it to date.

- *Affordances enable the proposal and evaluation of hypotheses during plan understanding.*

This theoretical tenet is both uncontroversial and supported in the literature, although few publications state it in these terms. For instance, Sindlar and Meyer (2010) report a system that uses logical reasoning about affordances to generate hypotheses about a BDI agent's intentions in a video game, but also uses numeric scores to evaluate them. In contrast, Freedman, Jung, and Zilberstein (2015) describe a probabilistic approach that ranks all candidate activities, using information about tool affordances for evaluation but not hypothesis generation. We encourage researchers who work in this area to be more explicit about the ways in which affordances guide their systems' decision making.

- *Affordances aid the proposal and evaluation of actions during plan generation.*

This postulate is also supported by publications in the area. One example comes from Ugur, Oztop, and Sahin (2011), who use learned object affordances during planning to propose candidate actions whose conditions match the current state, but not to evaluate them. In contrast, Boularias et al. (2015) use information about affordances, acquired by reinforcement learning, to evaluate alternative actions by comparing the values expected from their application.

- *Primitive affordances are learned inductively whereas complex affordances are learned analytically.*

Nearly all computational research in this arena has focused on acquiring primitive affordances and has relied exclusively on inductive methods, which is consistent with the first half of our claim. For instance, Kjellström, Romero, and Kragić (2010) describe a statistical approach to learning primitive affordances from observation for use in activity recognition, whereas Ugur et al. (2011) learn action models from exploration that map continuous features of objects to effect categories. Similarly, Boularias et al. (2015) report a system that estimates the expected values of actions in different situations, which they view as affordances, from delayed rewards. More interesting is recent work by Sridharan, Meadows, and Gomez (2017) that learns primitive affordances inductively and then combines them analytically into composite affordances on finding that sequences of actions achieve the agent's goals. However, this is the only work we have found that addresses the second half of our final tenet.

In summary, a number of theoretical claims about affordances appear to be novel, while others have received little attention. Taken together, they offer a new perspective that can drive work on embodied agents in interesting directions.

6 Concluding Remarks

In the preceding pages, we presented an account of affordances in intelligent systems. Our theory postulated these structures are reified concepts that specify when skills have particular effects for given agents, that allow graded membership, and that can be composed from more basic affordances. An intelligent system can use such structures to hypothesize and evaluate candidate plans that help understand others' behavior and achieve its own goals. Finally, such an agent can acquire affordance concepts from experience through a mixture of inductive and analytic learning mechanisms. We saw that others have explored some of these ideas, but that some appear novel, and there is no existing account of affordances that combines them into a unified theory.

In future research, we should incorporate these ideas into an implemented system, ideally an existing agent architecture that makes assumptions which are largely consistent with the new postulates (e.g., Li et al. 2012). We should also demonstrate the extended architecture on scenarios that illustrate the representation, use, and acquisition of graded, composite affordances for agents with different abilities. Finally, we should carry out experiments that test the benefits of affordance-driven processing over alternative approaches to intelligent systems. If studies reveal that this leads to better explanations, more effective plans, and reduced search, they will serve as evidence that supports the theory.

Acknowledgements

This research was supported by Grant N00014-17-1-2434 from ONR, which is not responsible for its contents.

References

Boularias, A.; Bagnell, J.; and Stentz, A. 2015. Learning to manipulate unknown objects in clutter by reinforcement. *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 1336-1342. Austin, TX: AAAI Press.

Choi, D. 2010. Nomination and prioritization of goals in a cognitive architecture. *Proceedings of the Tenth International Conference on Cognitive Modeling*, 25–30. Philadelphia, PA.

Ellis, R.; and Tucker, M. 2000. Micro-affordance: The potentiation of components of action by seen objects. *British Journal of Psychology* 91:451–471.

Freedman, R. G.; Jung, H.-T.; and Zilberstein, S. 2015. Temporal and object relations in unsupervised plan and activity recognition. *Proceedings of AAAI Fall Symposium on AI for Human-Robot Interaction*, 51–59. Arlington, VA: AAAI Press.

Horton, T. E.; Chakraborty, A.; and St. Amant, R. 2012. Affordances for robots: A brief survey. *Avant* 3:71–84.

Iba, G. A. 1989. A heuristic approach to the discovery of macro-operators. *Machine Learning* 3:285–317.

Gibson, J. J. 1977. The theory of affordances. In R. E. Shaw and J. Bransford (Eds.), *Perceiving, acting, and knowing*. Hillsdale, NJ: Lawrence Erlbaum.

Kjellström, H.; Romero, J.; and Kragić, D. 2011. Visual object-action recognition: Inferring object affordances from human demonstration. *Computer Vision and Image Understanding* 115:81–90.

Li, N.; Stracuzzi, D. J.; and Langley, P. 2012. Improving acquisition of teleoreactive logic programs through representation extension. *Advances in Cognitive Systems* 1:109–126.

Meadows, B.; Langley, P.; and Emery, M. 2014. An abductive approach to understanding social interactions. *Advances in Cognitive Systems* 3:87–106.

Sahin, E.; Cakmak, M.; Dogar, M. R.; Ugur, E.; and Ucoluk, G. 2007. To afford or not to afford: A new formalization of affordances toward affordance-based robot control. *Adaptive Behavior* 15:447–472.

Sarathy, V.; and Scheutz, M. 2016. A logic-based computational framework for inferring cognitive affordances. *IEEE Transactions on Cognitive and Developmental Systems*, *8*.

Shen, W-M.; and Simon, H. A. 1989. Rule creation and rule learning through environmental exploration. *Proceedings of the Eleventh International Joint Conference on Artificial intelligence*, 675–680. Detroit: Morgan Kaufmann.

Sindlar, M.; and Meyer, J.-J. 2010. Affordance-based intention recognition in virtual spatial environments. *Proceedings of the Thirteenth International Conference on Principles and Practice of Multi-Agent Systems*, 304–319. Kolkata, India.

Sridharan, M.; Meadows, B; Gomez, R. 2017. What can I not do? Towards an architecture for reasoning about and learning affordances. *Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling*, 461–469. Pittsburgh, PA: AAAI Press.

Stoffregen, T. A. 2003. Affordances as properties of the animal-environment system. *Ecological Psychology* 15: 115–134.

Ugur, E.; Oztop, E.; and Sahin, E. 2011. Goal emulation and planning in perceptual space using learned affordances. *Robotics and Autonomous Systems* 59:580–595.

Vera, A.; and Simon, H. A. 1993. Situated action: A symbolic interpretation. *Cognitive Science* 17:7-48.

Zech, P.; Haller, S.; Lakani, S. R.; Ridgeand, B.; Ugur, E.; and Piater, J. 2017. Computational models of affordance in robotics: A taxonomy and systematic classification. *Adaptive Behavior* 25:235–271.

Learning Planning Operators from Episodic Traces

David Ménager
Electrical Engineering & Computer Science
University of Kansas
Lawrence, KS 66045 USA
dhmenager@ku.edu

Dongkyu Choi
Aerospace Engineering
University of Kansas
Lawrence, KS 66045 USA
dongkyuc@ku.edu

Mark Roberts, David W. Aha
Naval Research Laboratory, Code 5514
Washington, DC, 20375 USA
mark.roberts@nrl.navy.mil
david.aha@nrl.navy.mil

Abstract

Learning is an important aspect of human intelligence. People learn from various aspects of their experience over time. We present an episodic infrastructure for learning in the context of a cognitive architecture, ICARUS. After a review of this architecture, we formally define the architectural extensions for episodic capabilities. We then demonstrate the extended system's capability to learn planning operators using the episodic traces from two Minecraft-like scenarios.

1 Introduction

Learning is of central importance to intelligent agents. From the beginning of artificial intelligence back in 1950's, researchers have recognized that the learning process is intimately tied to the nature of intelligence (Simon 1980). In order to adapt to dynamic environments, intelligent agents must possess mechanisms that allow them to acquire a broad repertoire of relevant behaviors. For this reason, there has been a significant amount of research on learning domain models in a variety of manners. But we rarely find any theories that provide a complete account of how experiences are gathered and how knowledge is derived from such experiences over time.

Our research aims to provide an infrastructure for organizing and processing collected experience, which then establishes a foundation for an experiential learning in intelligent agents. We model human *episodic* capabilities (Tulving 1983) in the context of a cognitive architecture, ICARUS (Langley and Choi 2006), and attempt to bridge these capabilities with other learning modalities. In this paper, we begin our study with the experiential learning of planning operators including action and event models. This will produce agents capable of learning throughout their lives to develop low-level expertise and adapt to dynamic environments. Such agents will also be able to recover from incorrect or incomplete knowledge over time. Additionally, because ICARUS learns structured models, agents retain the advantage of explainability.

Our work is motivated by situated agents that learn in changing, dynamic environments. Certainly, robots are one kind of such agent, but this paper focuses on a simulated domain described in the next section. After a description of this illustrative domain, we review the ICARUS architecture by providing necessary definitions that contextualize the episodic extensions we describe next. Then we present some preliminary results in the domain. Finally, we will discuss related work before we conclude.

2 Illustrative Domain

To motivate our research on episodic agents and evaluate our system's capabilities, we use a simplified version of a popular open-world game, Minecraft (Johnson et al. 2016), where players attempt to survive in a continuous, dynamic world by collecting resources, forging tools, building structures, and fighting enemies. Consider a novice agent learning from an expert player who starts at the lower left corner of a room. There are resources scattered around the room and a craft desk nearby the player. The player should gather the resources to make a sword for protection, but there are zombies in this room that guard the resources. The player must be careful because she will lose health if a zombie attacks her.

The expert player starts by selecting a resource and moving north toward it. Once she is on the same row as the resource, the player moves east toward it until she is on the same column. Now the player is standing by the resource and picks up the resource to hold it. But there was a zombie in the same location, so the player's health was reduced while the player was standing there. Then she moves south and then west to the craft desk. When the player arrives there, she puts down the resource on the desk. After repeating this process several times, the expert player would have gathered all the resources necessary to build a sword and achieve its mission by crafting one. The novice observer stores in its mind all the situations the expert has encoun-

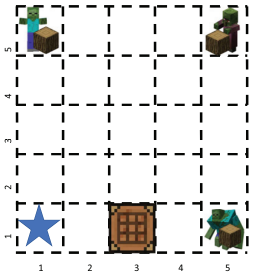


Figure 1: A 5x5 notional plot of Minicraft.

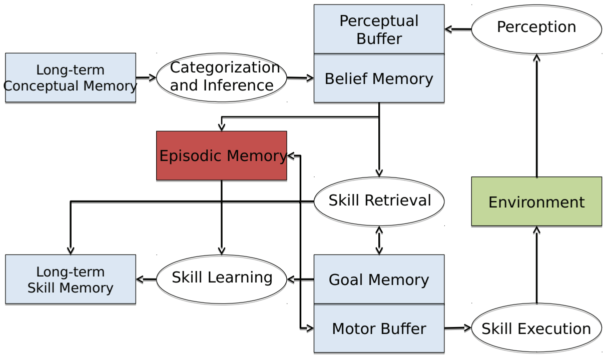


Figure 2: ICARUS cycle prior to episodic memory extension.

tered, and learns action and event models from them that it will be able to use to play the game.

We began our work by creating a grid world, *Minicraft*, that is inspired by the original Minecraft. Although simplified, this game captures enough dynamism to demonstrate the learning ability of our system. Figure 1 shows a notional view of Minicraft, which consists of four entities: resource, craftdesk, zombie, and the agent. The only entities with dynamic properties are the zombie and the agent. The agent begins at the star and moves one grid at a time while picking up or dropping resources and crafting items. Zombies, once placed on the map, are stationary, but provide dynamism to the world by decreasing the agent's health by one for every moment that the agent resides in the same grid as the zombie. All world dynamics, such as the effects of movement and action are unknown to the observer.

3 ICARUS Review

As a cognitive architecture, ICARUS provides a framework for modeling human cognition and programming intelligent agents. The architecture makes commitments to its representation of knowledge and structures, the memories that store these contents, and the processes that work over them. ICARUS shares some of these commitments with other architectures like Soar (Laird 2012) and ACT-R (Anderson and Lebiere 1998), but it also has distinct characteristics like the architectural commitment to hierarchical knowledge structures, teleoreactive execution, and goal reasoning capabilities (Choi 2011). Section 3.1 describes the key knowledge and memory structures of ICARUS, while Section 3.2 outlines how processes operate on these memories as part of a cognitive cycle.

ICARUS learns in the context of propositional states and action event models. Given a finite set of first order propositions P we define a propositional language $\mathcal{L}(P)$, and a finite set of labeled procedures, called *actions*, $\mathcal{A}$ such that $\mathcal{L}(P) \cap \mathcal{A} = \emptyset$.

3.1 Representation and Memories

ICARUS distinguishes two main types of knowledge: concepts and skills which represent semantic and procedural knowledge, respectively. Both have parameterized (i.e., lifted) variants that are grounded when variables are assigned to objects. Figure 2 shows the long-term and short-term memories of ICARUS, in which concepts and skills are stored. Paramaterized concept and skill definitions are stored in conceptual and procedural long-term memories, respectively. Instances of these definitions are stored in their respective conceptual or procedural short-term memories.

Concepts describe certain aspects of a situation in the environment. They resemble horn clauses (Horn 1951), complete with a predicate as the head, perceptual matching conditions, tests against matched variables, and references to any sub-relations.

Definition 1 (Concepts (C)) *A **primitive** concept is defined over P as $c_i = \langle \lambda, \epsilon \rangle$ where $\lambda \in P$ known as the concept head, ϵ denoting elements to pattern match in the world state S, where S is a subset of P. Let C_p be the set of primitive concepts. A **non-primitive** concept is defined over $P \cup C_p$ as $c_j = \langle \lambda, \epsilon, \gamma \rangle$ where γ denotes c_j's subrelations. We can further define non-primitive concepts over $P \cup C_p \cup C_n$, where C_n is the set of non-primitive concepts.*

Figure 3 shows example concepts for Minicraft. The first, `north-of`, is a primitive concept that describes the situation where a zombie is to the north of the agent, using perceptual matching and test conditions for *self* and *zombie*. The second, `on-horizontal-axis`, depicts a non-primitive concept where a zombie is on the same horizontal line as the agent. The third, `standing-by`, describes an even more abstract non-primitive concept where the zombie is standing right next to the agent.

Skills describe procedures to achieve certain concept instances in the environment. These are hierarchical versions of STRIPS operators (Fikes and Nilsson 1971) with a named head, perceptual matching conditions, preconditions that need to be true to execute, direct actions to perform in the

```
((north-of ?o1 ?self)
   :elements ((self ?self y ?y) (zombie ?o1 y ?y1))
   :tests ((> ?y1 ?y)))
((on-horizontal-axis ?o1 ?self)
   :elements ((self ?self) (zombie ?o1))
   :conditions ((not (north-of ?o1 ?self))
                (not (south-of ?o1 ?self))))
((standing-by ?self ?o1)
   :elements ((self ?self) (zombie ?o1))
   :conditions ((on-horizontal-axis ?o1 ?self)
                (on-vertical-axis ?o1 ?self)))
```

Figure 3: Three ICARUS concepts in the Minicraft domain.

```
((gather-resource ?o1)
   :elements ((self ?self) (resource ?o1))
   :conditions ((not (carrying ?any))
                (standing-by ?self ?o1))
   :effects ((carrying ?o1))
   :actions ((*pick-up-resource ?o1)))
((go-to ?o1)
   :elements ((self ?self))
   :conditions ((north-of ?o1 ?self))
   :subskills ((go-up-to ?o1))
   :effects ((standing-by ?self ?o1)))
((gather-resource ?o1)
   :elements ((self ?self) (resource ?o1))
   :conditions ((not (carrying ?any)))
   :subskills ((go-to ?o1) (gather-resource ?o1))
   :effects ((carrying ?o1)))
```

Figure 4: Three ICARUS skills in the Minicraft domain.

world or any sub-skills, and the intended effects of the execution.

Definition 2 (Skills (K)) *Given the finite set of actions $\mathcal{A}$, a skill defined over $C \cup S$ where C is the set of concepts and S is a propositional state, is a* primitive *skill if $k_i = \langle \epsilon, \gamma, \alpha, \sigma, \eta \rangle$, where pattern match conditions $\epsilon \subseteq S$, preconditions $\gamma \subseteq \{\lambda | \langle \lambda, \cdot \rangle \in C\}$, actions $\alpha \subseteq \mathcal{A}$, sub-skills $\sigma = \emptyset$, and effects $\eta \subseteq \{\lambda | \langle \lambda, \cdot \rangle \in C\}$. Let K_p be the set of primitive skills.*

A skill defined over $C \cup S \cup K_p$ is a non-primitive *skill if $k_j = \langle \epsilon, \gamma, \alpha, \sigma, \eta \rangle$, where $\epsilon \subseteq S, \gamma \subseteq \{\lambda | \langle \lambda, \cdot \rangle \in C\}, \alpha = \emptyset, \sigma \subseteq K_h$, and $\eta \subseteq \{\lambda | \langle \lambda, \cdot \rangle \in C\}$. K_h is the set of non-primitive skills.*

Figure 4 shows example skills for Minicraft. The first, `gather-resource`, is a primitive skill that describes a procedure to collect a resource that is executable when the agent is not carrying anything and is standing next to the resource. This skill uses a direct action to pick up the resource and its intended effect is carrying the resource. The bottom two are non-primitive skills that use sub-skills: `go-to` uses a sub-skill `go-up-to` to achieve the goal of standing near the object, while `gather-resource` uses the two sub-skills above it to collect a resource.

3.2 The ICARUS Cognitive Cycle

The ICARUS architecture operates in a cognitive cycle repeating two steps: conceptual inference and skill execution. *Conceptual inference* is the process of creating concept instances (i.e., beliefs). At the beginning of each cycle, the system receives sensory input from the environment as a list of objects with their attribute-value pairs; this can be thought of as the world state and is represented as propositions. Based on this information, the architecture infers the concept instances (i.e., beliefs) that are true in the current state by matching its concept definitions to perceived objects and other concept instances in a bottom-up fashion.

In summary, Figure 2 shows concept definitions housed in the conceptual long-term memory are used to infer the beliefs of the system from the world state and are stored as concept instances in the conceptual short-term memory.

Definition 3 (Beliefs (B)) *Let C be the set of concepts. $\forall c = \langle \lambda, \epsilon, \gamma, \tau \rangle \in C$, $\exists$* belief *$b = \langle \lambda, \epsilon, \gamma, \tau, \beta \rangle$ where β represents bindings that ground b on the perceptual elements, ϵ. Let B be the set of all possible beliefs, and let $\mathcal{B} = 2^B$ be the set of all* belief states. *A* belief state *$s \in \mathcal{B}$.*

Skill execution proceeds after conceptual inference whereby ICARUS finds all the relevant skill definitions for the current goal(s) that are executable based on the current beliefs. ICARUS chooses a skill and sets it as its *intention* and executes it in the world.

Definition 4 (Intentions (ι)) *Let K be the set of skills. $\forall k = \langle \epsilon, \gamma, \alpha, \sigma, \eta \rangle \in K$, there exists* intention *$\iota = \langle \epsilon, \gamma, \alpha, \sigma, \eta, \beta \rangle$ where β represents bindings that ground ι in the belief state.*

Each cycle may introduce changes in the environment, which may modify the sensory input for the next cycle, resulting in new beliefs and intentions. The architecture iterates in this manner until all of its goals are achieved or its operations are terminated for any other reasons.

4 Constructing Episodes

We now shift our attention to extending ICARUS with an episodic memory. In particular, we highlight the core data structures of ICARUS's Episodic Memory (Section 4.1), how it encodes episodes within that memory through a process called event segmentation (Section 4.2), and how it generalizes episodes over time (Section 4.3).

4.1 The Episodic Memory

The episodic memory in ICARUS is a long-term, cue-based memory that the agent uses to deliberately encode and retrieve episodes. The architecture organizes its episodic memory $E = \langle \rho, \mathcal{F}, \mathcal{T} \rangle$ in a compound structure composed of an episodic beliefs-action cache ρ, a concept frequency forest $\mathcal{F}$, and the episodic generalization tree $\mathcal{T}$.

Figure 5 shows how information is processed within the episodic memory and is discussed through this section. ρ acts as a storage for the agent's unprocessed history. We assume that the agent has sufficient memory to store the complete beliefs-action sequence. $\mathcal{F}$ records counts for the number of times concepts and their instantiations as beliefs have occurred during the execution of the agent. $\mathcal{T}$ is the main data structure that organizes and stores episodes; the contents of $\mathcal{T}$ are used in the process of learning new skills. The elements ρ and $\mathcal{F}$ (Definitions 5 and 6), discussed next, facilitate the workings of the event segmentation and episodic encoding (Section 4.2). Generalization with $\mathcal{T}$ is discussed in Section 4.3.

Since episodes are built on top of sequences of beliefs, we introduce first the beliefs-action cache, which stores the moment-by-moment changes in belief, inferred from the world state, as well as the actions that were taken based on those beliefs.

Definition 5 (Beliefs-action cache (ρ)) *The* beliefs-action cache ρ, *is an ordered sequence of belief-action pairs. This cache stores a complete, detailed history of what the agent observed. Figure 5 shows that the contents of the belief memory are inputs to the beliefs-action cache.*

Once these traces are collected, they must be processed for *interesting* events, which are tracked in the concept frequency forest.

Definition 6 (Concept frequency forest ($\mathcal{F}$)) *Let X be a set of location predicates, and let $Y = \{x.first | x \in S\}$ be the set of object types. A* concept frequency tree *is a tree whose the root μ is a location predicate from X. The children of μ are all the concepts the agent has observed in that location. For each child concept, c, of μ, $\exists$ a set of types from Y, to specify concept disjunctions. Under each disjunction, j, there exists concept instances. Each node in the tree has a count field, denoting the number of times this node has been observed. A* concept frequency forest *is a collection of concept frequency trees.*

ICARUS uses $\mathcal{F}$ to model *expectation violation*. The agent sets two thresholds: one for positive expectations and one for negated expectations. Any belief with a conditional probability, given the location, is greater than the positive threshold is said to be *expected*. Any belief with a conditional probability, given the location, is less than the negated threshold is *not expected* to be in the state. A belief that violates an expectation is a *significant belief*, which prompt the system to create an episode. This is a primitive method for novelty detection that only uses spatial information, but we can further extend the novelty detection method to include the temporal domain as well.

The episode structure defined in Definition 7 represents the agent's experiences in the architecture. Once they are stored in memory, episodes are processed to abstract general rules that allow the agent to predict environmental dynamics.

Definition 7 (Episode (ε)) *An episode is a tuple $\langle B_s, B_e, \Sigma, \psi \rangle$, where B_s is the start state of the episode, B_e is the end state of the episode, Σ is the set of significant beliefs in B_e, and ψ is a count for the number of times the episode has occurred.*

During episodic encoding, the start and final states are taken from the ρ (i.e., the beliefs-action cache). In the current implementation, B_s and B_e are consecutive belief states, but our work does not require this. Our rationale is psychologically inspired. When humans perform low-level actions, kicking a soccer ball for instance, humans know that the effect is not always observed in their next cognitive cycle. The ball travels in time before it reaches the goal. This dynamic is readily understood by most humans. Modeling actions with temporally delayed effects is part of our future work.

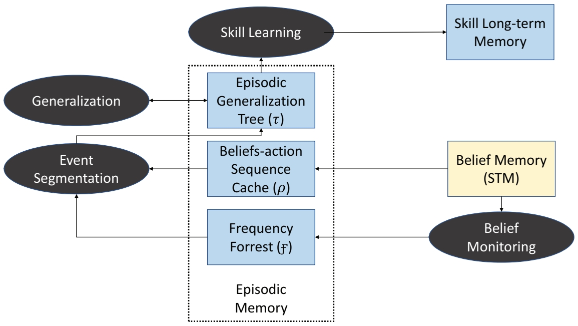


Figure 5: Block diagram depicting episodic memory components and information flow starting from the belief memory.

4.2 Episodic Encoding

Episodic Encoding in ICARUS is a two-step process. First, ICARUS operates on the ρ to returns a new episode ε. This is referred to as "Event Segmentation" in Figure 5. Once the episode exists, the second process places it into the episodic generalization tree. Algorithm 1 shows that encoding is triggered by the presence of one or more significant beliefs in belief state.

Algorithm 2 traces how episodes are inserted into the episodic generalization tree. Suppose the generalization tree contains several episodes. Γ is a list of sibling episodes under parent $\varrho \in \mathcal{T}$ If $\forall \varepsilon_i \in \Gamma, (\varepsilon_i, \varepsilon) \notin E$ then $(\varrho, \varepsilon) \in E$. That is ε becomes a child of ϱ. A new episode has successfully been encoded into the episodic memory. If $\exists \varepsilon_j \ni \varepsilon_j = \varepsilon$, then the counter for ε_j increments by one and ε is not inserted.

On every cycle, ICARUS records the belief state and executed actions into the episodic cache and updates $\mathcal{F}$. When the agent infers one or more significant beliefs, it encodes

Algorithm 1 CREATEEPISODE(ρ, loc, B_c)

```
$\rho$ is beliefs-action cache
$loc$ is current location
$B_c$ is current belief state
$B_{prev} \leftarrow$ last state in $\rho$
$\rho \leftarrow \rho.add(B_c, a)$
$sigs \leftarrow$ GETSIGNIFICANTBELIEFS($B_c, loc$)
if not NULL($sigs$) then
    $\varepsilon \leftarrow$ MAKEEPISODE($sigs, B_c, B_{prev}$)
    $\mathcal{T} \leftarrow$ INSERT($\varepsilon, \mathcal{T}$)
```

Algorithm 2 INSERTEPISODE($\varepsilon, \mathcal{T}$)

```
$queue \leftarrow \emptyset$
$temp \leftarrow$ root of $\mathcal{T}$
$match \leftarrow \emptyset$
$p \leftarrow \emptyset$
while not NULL($temp$) do
    $match \leftarrow$ STRUCTURALEQ?($temp$, $\varepsilon$)
    if $match$ is exact match then
        $temp.count \leftarrow temp.count + 1$
        Try to learn from $temp$ if count high enough
        BREAK
    else if $match$ is bc of unification then
        $temp.count \leftarrow temp.count + 1$
        $queue \leftarrow \emptyset$
        $queue \leftarrow temp$'s children
        Try to learn from $temp$ if count high enough
        $p \leftarrow temp$
    $temp \leftarrow queue$.FIRST
    $queue \leftarrow queue$.POP
if null($temp$) and $match$ not exact then
    $p \leftarrow p$.ADDCHILD($\varepsilon$)
    $\mathcal{T} \leftarrow$ GENERALIZE($p$, $\varepsilon$)
```

a new episode. The root node of the generalization tree is the most general episode and is allowed to have an arbitrary number of children. Under the root, episodes are grouped according to *structural similarity*. Two episodes e_1, e_2 are *structurally similar* if their significant beliefs unify. By "unify" we mean that there must exist a binding set that transforms the significant beliefs of e_1 to those of e_2 and vise versa. This is a rigid generalization scheme that needs more consideration in future work. Each child is a *k*-ary tree where $k \in \mathbb{N}$. Episodes become more specific at each decreasing level of the tree according to structural similarity. At the leaf nodes exist fully instantiated episodes.

4.3 Episodic Generalization

ICARUS supports generalization of the episodic tree during encoding of episode, ε_i. Definition 8 shows that an episode hierarchy is induced by structural similarity. Two sibling episodes $\varepsilon_i, \varepsilon_j$ generalize iff $\exists$ episode ε_g such that $(\varepsilon_g, \varepsilon_i) \in E$ and $(\varepsilon_g, \varepsilon_j) \in E$, but $(\varepsilon_i, \varepsilon_g) \notin E$ and $(\varepsilon_j, \varepsilon_g) \notin E$. This means that ε_g unifies with its children, $\varepsilon_i, \varepsilon_j$, but its children cannot unify with it because they contain more specified bindings. If ε_g exists, ICARUS tests to see if it is still more specific than the parent of ε_i. If so, then ε_g's parent becomes ε_i's parent and ε_g's children become $\varepsilon_i, \varepsilon_j$. The count for a generalized episode is the summation of the count of its children.

Definition 8 (Generalization tree ($\mathcal{T}$)) *An* episodic generalization tree *is a tree (V, E) where V is a set of episodes, and E is a set of edges. For any $\varepsilon_i, \varepsilon_j \in V, (v_i, v_j) \in E$ if they are structurally similar. An episode is said to be generalized or* partially instantiated *if the bindings contain one or more unbound variables.*

The generalization tree naturally lends itself to the learning process as a result of generalization. For example, if person x drops a glass on the ground and it breaks, and person y drops a glass on the ground and it breaks as well, ICARUS forms a generalized episode that implies if anyone drops a glass on the ground, it will break. The ability to gain knowledge in this way is central to general intelligence. As the tree adds more episodes, they are sorted into increasingly sensible taxonomies. The resulting tree after insertion is ICARUS' best estimate of the ideal generalization tree. This organizational structure was inspired by the incremental concept formation literature (Gennari, Langley, and Fisher 1989). As episodes become more general, the skills ICARUS learns from those episodes are equivalently general. So, generalizing skills is performed within the episodic generalization tree, not the skill learning algorithm.

5 Skill Learning using the Episodic Memory

In previous work, ICARUS supported learning by observing problem solving traces that include goals, conditions, and the skills used (Nejati 2011). The system relied on the explanations it generated based on the given trace, and this process required, at the very least, primitive skills in ICARUS' memory. In the current work, we start with only the concepts that are sufficient to describe situations in the world but the agent does not have any skills in its knowledge base.

ICARUS starts as an observer and records the history of belief states and ground actions in its episodic memory. As its experience accumulates, the agent will insert an episode whose count surpasses a predefined threshold for model learning. At that moment, the system uses the actions from $B_s \rightarrow B_e$ as a search cue for collecting other episodes where that ordering of actions took place. This trace of episodes is then used in the rule induction algorithm, MLEM2 (Grzymala-Busse and Rzasa 2010). Although we are using MLEM2, this need not be the case. Any rule learning algorithm may be used as long as there is a transformation from ICARUS's representation of experience to the representation that the learning algorithm requires. After learning, the agent can seamlessly utilize the learned skills during problem solving.

5.1 Learning Action and Event Models

In order to learn models of the world, ICARUS must first retrieve experiences via a retrieval cue. The system generates an observation, as defined in Definition 9 for each episode that matches the cue. For the case of model learning, the retrieval cue is some subset of actions a_i from $\mathcal{A}$. As the episodes are examined, matches are collected into an episodic trace of evidence related to a_i.

Definition 9 (Observations (O)) *Let $o = \langle s_i, a_i, s_f \rangle$ be an* observation *from ρ, the beliefs-action cache, where $s_i, s_f \in \varsigma$ are respectively initial and final belief states, and $a_i \subseteq \Lambda$ be the set of actions that transformed s_i to s_f An* episodic trace*, O is a collection of observations.*

MLEM2 learns rules from data tables, therefore, once the episodic trace is obtained it needs to be transform O into a table. The x-axis for this table is an enumeration of all the

Belief	ℓ_b	$\ell_b \cap \{1,3,4\}$
(holding sword1)	{1,2,3,4,5}	{1,3,4}
(holding nothing)	{6}	∅
(holding food1)	{7,8}	∅
(next-to ?zombie)	{1,3,4,7}	{1,3,4}
(next-to tree1)	{5,2, 6,8}	∅
(health good)	{1,2,3,4,5,6,7,8}	{1,3,4}

Table 1: Sample attribute and decision blocks.

unique beliefs in O, and the y-axis numbers each observation in O. Each belief, b on the x-axis has an associated list , $block_b = \{i | \langle s_j, a_j, s_k \rangle \in O[i], b \in s_j\}$ of the observation indices it appeared in. The last column of the data table is the list of the effects, fx for each associated observation. Table 1 summarizes the data table in a way that clearly shows each belief's block list. For example, the middle column states for the first row, that the (holding sword1) belief was present in observations 1 through 5.

For each effect, f in fx, the algorithm computes a list, $block_{fx} = \{i | \langle s_j, a_j, s_k \rangle \in O[i], f \in s_k\}$ of observation indices that it appeared in as well. MLEM2 tries to find, for each effect, conditions whose associated blocks cover the effect block. These coverings are what are the learned action and event models.

In this example, assume $a_i = ((*attack))$, and $fx = \{((zombie\text{-}dead\ ?zombie), \{1,3,4\}), ((wood\ wood1), \{2\})\}$.[1] MLEM2 attempts to find local coverings of fx from the list of belief conditions. MLEM2 tries the pair $(b, block_b)$ whose listing, $block_b$ intersected with an uncovered effect $block^0_{fx} = \{1,3,4\}$ is the largest. If $block_b \leq block^0_{fx}$, then that condition becomes a rule that covers that effect. If $block_c \nleq block^0_{fx}$ then other conditions need to be added to cover it. Once a rule has been found that covers all the cases of for an effect, the same process repeats for the uncovered effects in fx. In the example, the system learns the following rule: (next-to ?zombie) $\cap$ (holding sword) $\rightarrow$ (zombie-dead ?zombie).

In the ICARUS context, MLEM2 results are converted to action and event models, which are primitive skills. The left hand side of the rules become the preconditions, the right hand side would be the effects of the skill. The action information would capture what work needs to be done to realize the effects.

6 Experimental Setup

The goal with this research was to create an agent that could learn unknown domain dynamics from experience. Furthermore, we want a system that is flexible and continues learning over the course of its life to reflect the changes in the world's changing dynamics. We assume that the world is fully observable, and that the agent has a vocabulary that distinguishes belief states perfectly. Also, we assume effects come immediately after actions, and that the environment is not stochastic.

We tested on two scenarios. Each scenario has one expert with perfect concept and skill knowledge, and one observer with full observability of the state, perfect concept knowledge, but no skill knowledge (i.e., no knowledge of the domain dynamics). We are primarily interested in what action and event models the agent learns and know how they change in response to new evidence. In the first scenario, we place the expert at (1,1), and zombies and resources are at the other three corners. At (5, 1) there exists a craftdesk. The expert is tasked with collecting resources and placing them on the craftdesk. For the case of the expert, this problem is easily solved, but for the novice, we are interested in how well it learns the dynamics of the world. An example of an event model would be knowing that being next to a zombie reduces the agent's health, and an example of an action model would be learning about what happens to the state when the agent moves.

The second scenario extends the first with the zombies and resources have been randomly re-assigned to different corners. This makes for two different, but structurally identical scenarios. By doing this, we ensure that the agent constructs episodes that will generalize with the other episodes in its memory.

```
(achieve-bottom-horizontal-axis-and-more)
   :conditions ((at minicraft) (north-of r1 me)
            (north-of r3 me) (east-of r2 me)
            (east-of r3 me) (east-of craftdesk1 me)
            (north-of zombie2 me) (north-of zombie3 me)
            (east-of zombie1 me) (east-of zombie2 me)
            (good-health me) (on-ground r1)
            (on-ground r2) (on-ground r3)
            (on-vertical-axis r1 me)
            (on-vertical-axis zombie3 me)
            (on-horizontal-axis zombie1 me)
            (on-horizontal-axis craftdesk1 me)
            (on-horizontal-axis r2 me))
   :actions ((*move-up))
   :effects ((south-of ?r3 me) (south-of craftdesk1 me)
            (south-of ?zombie3 me)
            (bottom-of-horizontal ?zombie3)
            (bottom-of-horizontal ?r3)
            (bottom-of-horizontal craftdesk1))

(achieve-bottom-horizontal-axis-and-more)
   :conditions ((on-horizontal-axis ?r3 me)
               (on-horizontal-axis craftdesk1 me)
               (on-horizontal-axis ?zombie3 me))
   :actions ((*move-up))
   :effects ((south-of ?r3 me) (south-of craftdesk1 me)
            (south-of ?zombie3 me)
            (bottom-of-horizontal ?zombie3)
            (bottom-of-horizontal ?r3)
            (bottom-of-horizontal craftdesk1))
```

Figure 6: Learned action models for the `*move-up` action before (top) and after (bottom) generalization.

7 Results

We demonstrate that the agent is able to learn goal-directed, specific or generalized action and event models from experience. Because of the episodic memory, ICARUS agents have a mechanism for experiential learning which allows them to learn world dynamics in the form of ICARUS skills. The learned skills are continually revised according to evidence.

Figure 6 demonstrates how the action model for moving

```
(achieve-fair-health-and-more)
   :conditions ((good-health me) (on-ground r1)
                (healthy-standing-by zombie3))
   :actions (nil)
   :effects ((fair-health me) (slouching-by me zombie3))
```

```
(achieve-fair-health-and-more)
   :conditions ((good-health me)
                (healthy-standing-by ?zombie2))
   :actions (nil)
   :effects ((fair-health me) (slouching-by me ?zombie2))
```

Figure 7: Action models for the event model before learning (top) and after learning (bottom).

up changes with experience. The initial action model in Figure 6 (top) contains many irrelevant conditions, while the final version (bottom) contains no irrelevant conditions; not shown are intermediate versions. The same is true for the event model the agent learns for achieving (fair health). Figure 7 shows that the irrelevant condition is removed from the event model by the last refinement, where the event model also successfully generalizes the initial version (top) to the final version (bottom).

In our framework the system learns models based on the agent's interpretation of the ground truth. This is interesting because it clarifies certain properties of inference. Specifically, if an agent is lacking conceptual vocabulary to describe situations, its learned models will show evidence of stochasm. In other words, there will be cases where the same action occurred in identical belief states resulting in different effects.

8 Related Work

Earlier research in action recognition and learning aims to teach robots to recognize and perform human gestures (Yang, Xu, and Chen 1997). In that work the researchers used a discrete hidden Markov model to decode human intentions, and to learn the motor actions that controlled making gestures. Along this line, Liu et al. (2017) recently developed a multi-task learning system that hierarchically recognizes human actions. Also, another recent approach attempted to learn control policies for continuous, non-Gaussian stochastic domains (Wang et al. 2017). The work describes a reinforcement learning system that learns an incomplete policy for a discrete controller. Given the policy, a robot executes the action for the nearest state to the current one.

The main distinction from our work and these is that they do not learn action models in the way that we have defined them. The action models these systems learn are often limited to scenario-specific transition functions, and control policies. The semantic meaning of actions, however is still unknown to the agent, so planning with the notion of explicit goals is not possible. Moreover, when these system refer to action models they typically refer to modeling the human motor controls that produce gestures.

In addition to machine learning, researchers are also trying to learn operator descriptions that can be used in performance systems. As Langley and Simon point out, our goal is to understand and characterize the invariants of intelligence. Building systems that help explain how novices become experts in general is key to this endeavor. Wang et al. (1994) created a system built on PRODIGY (Carbonell et al. 1991) that incrementally learned planning operators based on STRIPS (Fikes and Nilsson 1971) via observation and practice. Expert demonstrations allowed the system to estimate initial versions of the operators. The agent refined its knowledge base by attempting to use learned operators to solve problems. The system was able to learn subgoal orderings for the operators, but the system could not learn operator decompositions, so operators were learned and stored in a flat structure. Gil et al. (1994) discussed how imperfections in domain knowledge do not always lead to planning or execution failures. They also presented a system that learns to refine imperfect operators by experimenting. The experimentation process can refine both operator pre and post conditions.

Another system, ALPINE provided methods for inducing abstraction hierarchies over operators (Knoblock 1990). Given a set of low-level operators, the system could induce abstraction hierarchies that reduced the search space.

Another interesting approach learned operators with associated numeric attributes to denote the utility of a particular operator (García-Martínez and Borrajo 2000). In this way the system favored more accurate operators. Walsh and Littman (2008) addressed the problem of efficiently learning STRIPS-like operators via experience. They define their own notion of an episode to be an initial state, s_0 goal state, and all state-action pairs following s_0 until the problem is solved or marked unsolvable. Their notion of episode, however, is not tied to a larger theory of episodic memory.

Lastly,Molineaux and Aha (2014) describe a surprise-driven method for learning event models. Given a problem, the system returned a plan of actions that would achieve the goal as well as a sequence of expected state changes caused by executing those actions. The system notices surprises when discrepancies exist between actual and expected state transitions. Discrepancies trigger an explanation module, DISCOVERHISTORY to hypothesize the cause of the discrepancies. When explanations fail, the system uses a variant of FOIL to learn an action model that repairs broken explanations.

In our work we addressed the problem of model learning from the vantage point of episodic memory for intelligent agents. Other research has investigated episodic memory. In the work most similar to ours, Nuxoll and Laird (2007) extended the Soar architecture (Laird, Newell, and Rosenbloom 1987) with episodic memory. They present results for action modeling in their work, but details about the learning mechanism are left out. There are also significant theoretical differences between the episodic memory in ICARUS and Soar. ICARUS has strong commitments to hierarchical organization of knowledge throughout the architecture, which helps support our theory for incremental learning. Soar, although it has had many successes, does not have such strict commitments to hierarchy. In their architecture episodes are stored in a flat container for experiences. Moreover, episodes in ICARUS have temporal components, meaning that they contain a sequence of states, whereas Soar's episodes do not have any temporal dimension.

9 Conclusion

We presented a new extension to the ICARUS architecture that allows agents to learn goal-directed planning operators from episodic traces. Our results from the Minicraft domain showed that our theory incrementally learns skills in a specific-to-general manner, and also refines skills based on evidence. This evidence is collected from ICARUS episodic memory, a dedicated facility for constructing, storing and organizing experience.

Acknowledgments

The research presented in this paper was supported in part by the Naval Research Laboratory under contract number N00173-17-P-0837 and the Naval Research Enterprise Internship Program.

References

Anderson, J. R., and Lebiere, C. 1998. *The atomic components of thought*. Mahwah, NJ: Erlbaum.

Carbonell, J.; Etzioni, O.; Gil, Y.; Joseph, R.; Knoblock, C.; Minton, S.; and Veloso, M. 1991. Prodigy: An integrated architecture for planning and learning. *ACM SIGART Bulletin* 2(4):51–55.

Choi, D. 2011. Reactive goal management in a cognitive architecture. *Cognitive Systems Research* 12:293–308.

Fikes, R., and Nilsson, N. 1971. STRIPS: a new approach to the application of theorem proving to problem solving. *Artificial Intelligence* 2:189–208.

García-Martínez, R., and Borrajo, D. 2000. An integrated approach of learning, planning, and execution. *Journal of Intelligent and Robotic Systems* 29(1):47–78.

Gennari, J. H.; Langley, P.; and Fisher, D. 1989. Models of incremental concept formation. *Artificial intelligence* 40(1-3):11–61.

Gil, Y. 1994. Learning by experimentation: Incremental refinement of incomplete planning domains. In *International Conference on Machine Learning*, 87–95.

Grzymala-Busse, J. W., and Rzasa, W. 2010. A local version of the mlem2 algorithm for rule induction. *Fundamenta Informaticae* 100(1-4):99–116.

Horn, A. 1951. On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic* 16(1):14–21.

Johnson, M.; Hofmann, K.; Hutton, T.; and Bignell, D. 2016. The malmo platform for artificial intelligence experimentation. In *IJCAI*, 4246–4247.

Knoblock, C. A. 1990. Learning abstraction hierarchies for problem solving. In *AAAI*, 923–928.

Laird, J. E.; Newell, A.; and Rosenbloom, P. S. 1987. Soar: An architecture for general intelligence. *Artificial Intelligence* 33(1):1–64.

Laird, J. E. 2012. *The Soar Cognitive Architecture*. Cambridge, MA: MIT Press.

Langley, P., and Choi, D. 2006. A unified cognitive architecture for physical agents. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*.

Langley, P. 1983. Learning search strategies through discrimination. *International Journal of Man-Machine Studies* 18(6):513–541.

Liu, A.-A.; Su, Y.-T.; Nie, W.-Z.; and Kankanhalli, M. 2017. Hierarchical clustering multi-task learning for joint human action grouping and recognition. *IEEE transactions on pattern analysis and machine intelligence* 39(1):102–114.

Molineaux, M., and Aha, D. W. 2014. Learning unknown event models. In *AAAI*, 395–401.

Nejati, N. 2011. *Analytical Goal-Driven Learning of Procedural Knowledge by Observation*. Ph.D. Dissertation, Stanford University.

Nuxoll, A. M., and Laird, J. E. 2007. Extending cognitive architecture with episodic memory. In *Proceedings of the Twenty-Second National Conference on Artificial Intelligence*, 1560–1565.

Simon, H. A. 1980. Cognitive science: The newest science of the artificial. *Cognitive science* 4(1):33–46.

Tulving, E. 1983. Elements of episodic memory.

Walsh, T. J., and Littman, M. L. 2008. Efficient learning of action schemas and web-service descriptions. In *AAAI*, volume 8, 714–719.

Wang, Z.; Jegelka, S.; Kaelbling, L. P.; and Lozano-Pérez, T. 2017. Focused model-learning and planning for non-gaussian continuous state-action systems. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, 3754–3761. IEEE.

Wang, X. 1994. Learning planning operators by observation and practice. In *AAAI*, 335–340.

Yang, J.; Xu, Y.; and Chen, C. S. 1997. Human action learning via hidden markov model. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 27(1):34–44.

Human-Agent Teaming as a Common Problem for Goal Reasoning

Matthew Molineaux,[1] **Michael W. Floyd,**[1] **Dustin Dannenhauer,**[2] **David W. Aha**[3]
[1]Knexus Research Corporation; Springfield, VA | {first.last}@knexusresearch.com
[2]NRC Postdoctoral Fellow; NRL; Navy Center for Applied Research in AI; Washington, DC | dustin.dannenhauer.ctr@nrl.navy.mil
[3]Naval Research Laboratory; Navy Center for Applied Research in AI; Washington, DC | david.aha@nrl.navy.mil

Abstract

Human-agent teaming is a difficult yet relevant problem domain to which many goal reasoning systems are well suited, due to their ability to accept outside direction and (relatively) human-understandable internal state. We propose a formal model, and multiple variations on a multi-agent problem, to clarify and unify research in goal reasoning. We describe examples of these concepts, and propose standard evaluation methods for goal reasoning agents that act as a member of a team or on behalf of a supervisor.

1 Introduction

An important focus of research on intelligent agents is to achieve goals quickly and reliably. In recent years, goal reasoning researchers have considered the issue of *goal change*, a process by which an agent can shift the overall focus of its activities. This change can be prompted by a nameless outside goal source and/or an internal motivation model. In this work, we advocate modeling the other agents whose goals an agent attempts to achieve. With this model change, it becomes clear that goal reasoning agents are particularly well-suited to being team players. We define a human-agent teaming model and problem, and discuss how future goal reasoning research can leverage it.

Research on goal reasoning has investigated multiple framework abstractions for algorithms and agent architectures (e.g., Goal-Driven Autonomy (Molineaux, Klenk, and Aha 2010) and the Goal Lifecycle (Roberts et al. 2014)), but has not focused on common problems. Areas such as reinforcement learning and automated planning have benefited greatly from such a focus, receiving additional attention from competitions and comparing results via easy-to-use benchmarks. While one problem may not suffice to compare all goal reasoning agents, a small number of common problems could facilitate comparative publications, and thereby focus goal reasoning research. This paper focuses on elaborating this position, and a candidate formal framework for describing classes of problems; we expect that future work will specify concrete representations and initial problems.

In Section 2, we provide a formal description of a general *human-agent teaming* problem, along with several important variations that are commonly encountered in goal reasoning research. We then discuss some examples of the concepts described in Section 3, and discuss useful metrics for comparison in Section 4. Finally, in Section 5 we conclude.

2 Models of Goal Reasoning for Human-Agent Teaming

In recent work, goal reasoning systems have explicitly reasoned over the presence of other agents and their goals. For example, goal reasoning agents may be aware that their opponent in a real-time strategy game is attempting to defeat them (Weber, Mateas, and Jhala 2010; Jaidee, Muñoz-Avila, and Aha 2013; Dannenhauer and Muñoz-Avila 2015), that other agents may attack them (Bonnano et al. 2016), or that other agents may impede them (Cox 2013). Other work has described explicit exchange of goals and other information between agents and humans for the purpose of general collaborative tasks (Geib et al. 2016), control of unmanned vehicles (Richards and Stedmon 2017), and autonomous community formation (Golpayegani and Clarke 2016). The framework presented here is designed to facilitate communication and comparison of agents that work together in these ways. Concepts described here help with the modeling of the goals, plans, and motivations of other agents, especially those that reason over goals themselves. In the spirit of the successful reinforcement learning problem (Sutton and Barto 1998), we describe a simple set of functions and informational items intended to be general enough to be easily applied and used by all agents that solve these problems. In order to keep this framework generic and approachable, we avoid committing to representations and functions that many agents may not be able to provide.

In our model (Figure 1), a team is situated in an environment. This team can comprise goal reasoning agents, human teammates, and other software agents. At each time t ($t \in T$, the set of discrete time points at which communications occur), each *teammate* observes the environment. The environment's state is given by s_t ($s_t \in S$, the set of all environment states), and teammate m ($m \in M$, the set of teammates) receives an observation o_t^m ($o_t^m \in O$, the set of all observations). The environment creates individualized observations for each agent; we model the observation generation process as a function $obs^m : S \rightarrow O$. Teammates can perform

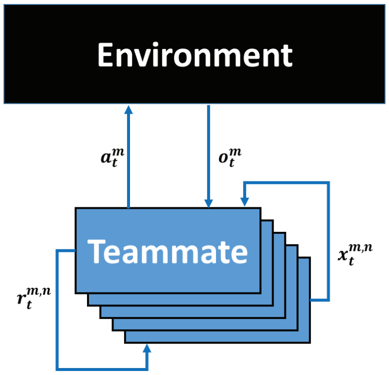


Figure 1: Human-Agent Teaming Problem Model

an action at each time t, denoted a_t^m ($a_t^m \in A$, the set of all actions). Changes in the environment are dependent on these actions as well as the prior state, which we model as the transition function $\lambda : S \times A^{|M|} \rightarrow S$. This generic representation allows for description of a wide variety of environments, including those with heterogeneous observability, exogenous events, and role-based actions; however, it does not permit continuous time.

Acting as teammates imposes some extra requirements on an agent. Work in human factors (Klein et al. 2004) has recognized four distinct requirements for acting as a member of team. Loosely summarized, they are: (1) agree on common goals; (2) direct and take direction from other teammates; (3) predict the behavior of other teammates and act in a way they can predict; and (4) maintain a common understanding of the shared environment. To support these requirements, in our framework teammates communicate via requests and explanations. A teammate m can make a *request* $r_t^{m,n}$ of another teammate $n \in M$ ($r_t^{m,n} \in R$, the set of all requests). Requests should describe everything agent m desires of agent n at time t. They are used both for direction and describing desired changes to common goals. Our model makes no specific commitment to representation; however, we expect that goal reasoning agents might directly exchange lists of goals, preferences, and constraints.

Explanations are intended to communicate information about an agent's internal state that motivates that agent's current behavior (e.g., "I moved the box because it was blocking my vision", "My battery is low so my movement range is limited"). Each teammate m provides an *explanation* $x_t^{m,n}$ to each other teammate n ($x_t^{m,n} \in X$, the set of all explanations). These explanations should help other teammates to understand an agent's actions and predict their future actions, to facilitate coordination. One particular area of importance is that an agent should explain why it does or does not pursue another agent's request; if an agent does not, for example, have sufficient resources to succeed, this may prompt the requester to provide resources or assistance.

Note that the explanations described here are proactive and not query-based. While query-based explanations are an important problem, a clean separation of agent-based coordination and decision-making issues from natural-language issues will permit objective evaluations and comparisons without human interaction issues. We expect, however, that an external query interface could be provided that translates queries into informational requests.

Each teammate m uses the various pieces of information they have received over time[1] (i.e., observed environment states, received requests, and received explanations) along with their sent requests (and, implicitly, their internal motivations) to guide their action selection policy $\pi^m : O^{|T|} \times R^{|M|} \times X^{|M| \times |T|} \times R^{|M|} \rightarrow A$. This policy is expected to be dynamic, and may be influenced by an agent's interactions with its teammates, as well as by the environment. A typical goal reasoning agent's policy may involve considering and reselecting goals and replanning to achieve them, but the model accommodates various types of policies.

We also model the *satisfaction* of each teammate, which describes how well an agent's desires are being met. Satisfaction is a function of an agent's observations (which may indicate the achievement of desired states), requests made and received (which help determine the success and failure of collaboration), and explanations received (which may justify failures or provide confidence in the current collaboration): $sat^m : O^{|T|} \times R^{|M|} \times X^{|M| \times |T|} \times R^{|M|} \rightarrow \mathbb{R}$. The satisfaction of the entire team can also be modelled as a function of each teammate's satisfaction ($f(sat^1, \ldots, sat^{|M|})$); optimizing this measure incorporates an agent's own satisfaction, as well as the estimated satisfaction of each of its $|M| - 1$ teammates.

To exemplify how our model could be used in practice, we describe it in terms of four variations on the human-agent teaming problem that describe existing goal reasoning work: *single supervisor*, *silent teammates*, *silent assistant*, and *rebel agent*. These examples are not meant to be exhaustive, but instead to show that our model can represent common team structures encountered in goal reasoning research.

2.1 Single Supervisor

Even autonomous goal reasoning agents often receive goals or tasks from an outside source. In this framework, we model that source as an agent who makes requests and wants explanations to understand what the agent is doing to fulfill them. This results in the Single Supervisor version of the human-agent teaming problem model, shown in Figure 2. In this version, an agent has a single teammate whose satisfaction it wishes to maximize, referred to as the *supervisor*. While both teammates can sense and act in the environment[2], the superior-subordinate relationship results in requests and explanations being unidirectional (i.e., the agent cannot make requests of the supervisor and the supervisor does not explain itself to the agent). As such, the agent's action selection policy does not include explanations it has received or

[1] We assume that, since the requests at the current time contain the complete request to/from each agent, the policy does not need to consider past requests. If this is not the case, the action selection policy can be extended to include past requests.

[2] Although the supervisor does not need to be situated in the environment.

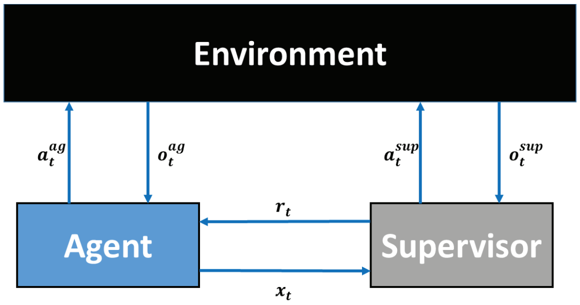


Figure 2: Single Supervisor version of the Human-Agent Teaming Problem Model

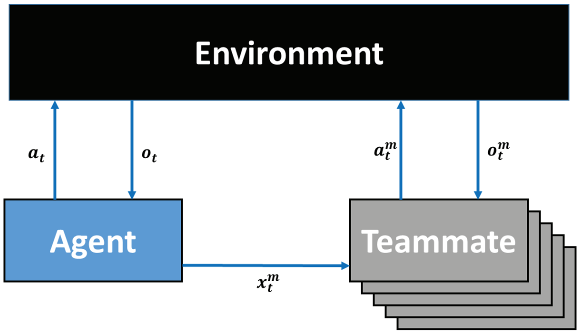


Figure 3: Silent Teammates version of the Human-Agent Teaming Problem Model

requests it has sent, and only deals with a single teammate (the policy is simplified to $\pi : O^{|T|} \times R \rightarrow A$). The primary performance measure for this problem is the supervisor's true satisfaction, measured either at the termination of interaction, or as an average over time.

2.2 Silent Teammates

In the Silent Teammates version of the human-agent teaming problem model (Figure 3), an agent operates as a member of a human-agent team, but does not receive any direct requests from its teammates. This is an unusual teaming arrangement, but necessary when a team is communication-restricted in some way (possibly to avoid giving an adversary knowledge). In this problem, the agent does not make requests of other teammates, nor expect explanations from them. However, the agent still provides an explanation on demand, to assist teammates in understanding when they have questions. An example of such a goal reasoning agent is the Autonomous Squad Member (ASM), an agent controlling an unmanned ground vehicle that is embedded in a team of humans (Gillespie et al. 2015). The ASM agent must infer and respond to teammates' desires (e.g., follow along, provide cover in a fight) without explicit requests. This results in an action selection policy that inputs only observations: $\pi : O^{|T|} \rightarrow A$. Similarly, the satisfaction function does not include requests: $sat^m : O^{|T|} \times X^{|M| \times |T|} \rightarrow \mathbb{R}$. The primary performance measure in this problem is the team's overall satisfaction.

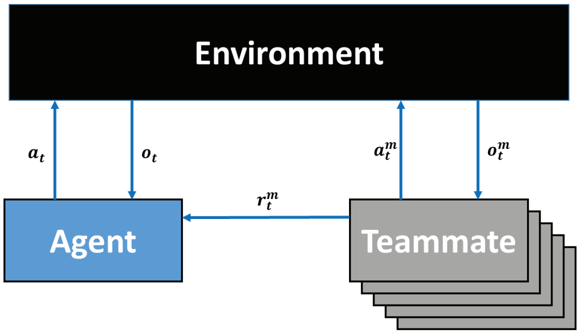


Figure 4: Silent Assistant version of the Human-Agent Teaming Problem Model

2.3 Silent Assistant

The Silent Assistant version is a multi-agent teaming problem with no explanation requirement (Figure 4). In this example, the agent assists one or more other agents by acting on their requests, but does not provide explanations, receive explanations, or make requests of others (i.e., it does not initiate coordination). An example of such a goal reasoning agent is the Tactical Battle Manager (TBM), an agent that controls an unmanned air vehicle while serving as a wingman for an aircraft controlled by a human pilot (Floyd et al. 2017). The TBM operates autonomously but receives explicit tasks from a human pilot. The lack of communication from the agent is largely due to the real-time adversarial nature of the domain; goal changes are motivated by dangerous situations or opportunistic targets, so explanations are not a primary requirement for this system. Additionally, since the TBM is a human pilot's wingman, it serves a subordinate role and therefore does not generate requests. As such, the agent's action selection function and the satisfaction functions do not include explanations or requests from the agent ($\pi : O^{|T|} \times R^{|M|} \rightarrow A$, $sat^m : O^{|T|} \times R^{|M|} \rightarrow \mathbb{R}$). The primary performance measure is the team satisfaction function.

2.4 Rebel Agent

The previous three problem versions we described assume that the agent's primary drive is to satisfy teammates' requests. In the Rebel Agent version (Coman, Gillespie, and Muñoz-Avila 2015), an agent has internal goals or motivations that differ from (and may conflict with) those of its teammates. There are two ways in which a rebel agent can be represented using our model. The simplest method is to consider the agent as a member of its team but having internal motivations that are unknown to its teammates. Thus, when attempting to maximize team satisfaction it may prioritize its own satisfaction above the satisfaction of its teammates (e.g., provide them with different weights). The ARTUE agent (Molineaux, Klenk, and Aha 2010) is a rebel agent that receives explicit requests in the form of goals that it may choose to ignore in order to achieve goals more important to it. A more complex representation would be to consider the agent to be a member of two teams concurrently (i.e., in Figure 1 the agent would be at the intersection of two teams). For example, consider an agent that is a member of a *corporate catering team*, but is also a member of a *vegetarian team*. While the agent contributes toward achieving catering goals (e.g., host a successful event, maximize profit) it may choose actions to maximize the vegetarian team's satisfaction (e.g., minimize the amount of meat used). In the internally motivated case, the primary performance measure is a team/rebel satisfaction function $f(sat^1, \ldots, sat^{|M|}, mot(s_{final}))$, where $mot(s_{final})$ describes how well a rebel's internal motivations are satisfied in the true final state of the environment. In the dual-membership case, the primary performance measure is a combined function of two (or potentially more) team satisfaction functions: $f_C(f_1(sat^1, \ldots, sat^{|M|}), f_2(sat^1, \ldots, sat^{|M|}))$.

2.5 Assumptions

Consideration of important assumptions is necessary for this framework. Existence of the transition and observation functions means that environments can be static or dynamic, deterministic or probabilistic, and fully or partially observable. Existence of the policy and satisfaction functions of teammates implies that we should also consider whether to assume complete or incomplete knowledge about these functions, and whether information given regarding them (i.e., requests and explanations) is perfect or noisy. This cuts across all problems, and those purporting to address these problems should state their assumptions regarding these functions.

3 Examples

Requests and explanations can take many forms including natural language utterances, structured text, or low-level state representations. In this section we provide examples of requests, explanations, and how they can be used.

Requests: In general, we expect requests to vary in complexity across agents. An example complex request representation might be a tuple $\langle S_{avoid}, F_{prefs}, G, C\rangle$, including constraints $S_{avoid} \subset S$ in the form of states to avoid (e.g., "battery should never fall below 10%"), preference functions $F_{prefs} : S \times S \rightarrow \{True, False\}$ (e.g., "spend as little money as possible"), goal states $G \subset S$ (with or without priorities), and context C that describes why achievement of a particular goal is desired (e.g., the reason for requesting an agent to cook food could be because (1) 'supervisor is hungry' or (2) 'supervisor needs to bring food to a dinner party later'). Context and preferences are especially relevant for goal reasoning agents, as these can guide which goals should be considered when goal change is warranted. Additionally, the reasons for a supervisor's request of a goal are likely to be useful in making goal change decisions; for example, the context may include a higher-level goal of which the current request is a subgoal (e.g., a "cook food" goal is a subgoal of a ¬hungry goal).

Explanations: An important reason for explanations is that goal reasoning agents may change their local objectives (i.e., subgoals) in response to changes in the environment preventing the accomplishment of the original task. Thus, whenever an agent changes its goal, an explanation could be a tuple $\langle g_{failed}, c_{failed}, g_{new}, p_{new}\rangle$ composed of a failed goal g_{failed}, description of state properties that prevent goal achievement c_{failed}, new goal g_{new}, and new plan p_{new}. Note that in this framework, explanations are always proactive for simplicity of discussion; to support reactive explanations, an external interface could store this information to present to a human in answer to specific queries.

A Supervisor Requests Cake: We now describe an example of the Single Supervisor problem: first, a human supervisor σ makes a request of a chef agent α to "*bake me a chocolate cake that I can eat when I get home*". Here, the request $r_t^{\sigma,\alpha}$ is the tuple $\langle\emptyset, \emptyset, \{\{$*exists(chocolate-cake), on(chocolate-cake,table)*$\}\}, \{$*hungry(me), wants(me, chocolate)*$\}\rangle$, which describes a single goal state based on the original English utterance (translating human utterances to goals has garnered attention in the human-robot interaction community, see (Briggs, McConnell, and Scheutz 2015) for an example). No constraints or preferences are provided.

The chef agent α represents its supervisor's satisfaction function sat^{σ} as a weighted average of (1) the percentage of his desires that are satisfied in the current state and (2) the time delay between t (time of request issuance) and t_a (time of request achievement). Based on this, the agent uses an automated planner to produce a plan that achieves the requested goal in the shortest possible time. Its policy π^{α} removes the first action from this plan and executes it; this is repeated until the following action a_t^{α} is known to be inadmissible based on a state observation o_t^{α}. We now describe a situation that may warrant the agent to consider goal change.

Soon after it begins acting to achieve the goal, the agent discovers it cannot continue baking because there is no cake flour in the kitchen. The agent considers adoption of a new goal *acquire(cake-flour)*, and creates a plan: go to the grocery store, purchase cake flour, and return. However, the plan to accomplish the new goal would significantly increase the time required to fulfill the supervisor's request. Knowing that the supervisor is hungry and wants chocolate cake, the chef agent decides to instead switch to a goal to make chocolate chip pancakes, which seems like a reasonable substitute. When the supervisor comes home, the agent provides him with an explanation:

$\langle\{$*exists(chocolate-cake), on(chocolate-cake, table)*$\}$,
$\{$*available(cake-flour)*$\}$,
$\{$*exists(pancakes), on(pancakes, table)*$\}$,
$\{$*acquire(pancake-mix), acquire(chocolate-chips),*
bake(pancakes, pancake-mix, chocolate-chips),
serve(pancakes)$\}\rangle$.

This explanation serves to communicate why the agent changed its goal, and what it did instead. If the context of the supervisor's request had been a birthday party, the agent α might have reasoned that the subgoal of going to the grocery store was warranted.

In general, the issue of how much information must be exchanged between teammates is unresolved. In this example, we assume sufficient knowledge to minimize the need for communication; for example, the agent knows that the supervisor's desires would be met to some degree by chocolate chip pancakes. Future work on goal reasoning agents will need to consider this question.

4 Evaluating Explainable Goal Reasoning Agents

We expect that typical evaluations will consider a specific problem and assumptions, and show results on a primary performance metric in a subset of domains. Results should be directly comparable with other agents that make the same assumptions, use the same domain, and use a similar set of teammates. For this reason, sharing domains as well as appropriate automated teammates (i.e., other software agents that are part of the team) should promote comparison.

When discussing the four versions of the human-agent teaming problem, we briefly described the various metrics that can be used to measure whether the goal reasoning agent is an effective member of the team. However, in addition to agent performance there is also the issue of how well the agent interacts with its human teammates. In these cases, evaluations should consider whether the provided explanations are appropriate for aiding human collaboration. We consider metrics for explanation as falling into four categories: *tests of explanation quality*, *tests of user satisfaction*, *tests of user comprehension*, and *tests of user or user-system team performance*. These are based directly on Hoffman, Klein and Mueller's (2017) work on evaluating explanations. Two agents need not use the same explanation representation (e.g., natural language, internal state variables) to be compared.

Tests of Explanation Quality: Experiments that measure explanation quality can be conducted without humans in the loop, but often still require a human to assess the results. These can be compared against explanations generated by another system or by a human. Some measures of explanation quality are surveyed in Table 1.

Tests of User Satisfaction: These should solicit a user's subjective satisfaction with an agent's performance, typically using Likert scale questions.

Tests of User Comprehension: These gauge how well explanations generated by an agent improve the accuracy of a user's mental model of an agent's behavior. For explainable autonomous agents, experiments could include questions about the system's policy to measure user understanding.

Table 1: Abstract Measures of Explanation Quality

Soundness	Plausibility, internal consistency
Appropriate Detail	Amount of detail and its focus points
Veridicality	Does not contradict the ideal model (although there are times when inaccurate explanations work better for some users and some purposes)
Usefulness	Fidelity to the designer's or user's goal for system use
Clarity	Understandability
Completeness	Relative to an ideal model
Observability	Explains an agent mechanism
Dimensions of Variation	Reveals boundary conditions

Tests of User or User-System Team Performance: These measure how explanation affects the user's ability to accomplish some task, often an interactive task involving the explaining agent. A scenario-specific performance metric can be used to evaluate the team's performance for this purpose. To provide a comparison, the same evaluation should be applied with and without agent-provided explanations, and, if possible, against a human-only team.

5 Conclusions and Future Work

We have presented new formal models and problem variations for human-agent teaming, in hopes of promoting comparisons, competitions, and sharing of evaluation code among goal reasoning researchers. We have made the case that explanation is an important and attainable capability for goal reasoning agents. Finally, we have described useful evaluations to be used to provide evidence of how well both goal reasoning agents and human-agent teams, perform.

In future work, we will produce refined models based on community feedback; furthermore, we will provide concrete problem instances and representations for use in benchmarking and comparison.

6 Acknowledgements

This research was developed with funding from DARPA. The views, opinions, and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the DoD or the U.S. Government. Table 1 is used with permission from the Institute for Human and Machine Cognition and was created by IHMC, MacroCognition LLC, and Michigan Technical University, with the support of the DARPA XAI Program DARPA-BAA-16-53. All rights are reserved by IHMC.

References

Bonnano, D.; Roberts, M.; Smith, L.; and Aha, D. 2016. Selecting subgoals using deep learning in Minecraft: A preliminary report. In *Working Notes of the IJCAI-16 Workshop on Deep Learning and Artificial Intelligence*.

Briggs, G.; McConnell, I.; and Scheutz, M. 2015. When robots object: Evidence for the utility of verbal, but not necessarily spoken protest. In *International Conference on Social Robotics*, 83–92. Springer.

Coman, A.; Gillespie, K.; and Muñoz-Avila, H. 2015. Case-based local and global percept processing for rebel agents. In *Case-Based Agents: Papers from the ICCBR 2015 Workshops*, 23–32.

Cox, M. T. 2013. Goal-driven autonomy and question-based problem recognition. In *Proceedings of the Second Annual Conference on Advances in Cognitive Systems, Poster Collection*, 29–45.

Dannenhauer, D., and Muñoz-Avila, H. 2015. Goal-driven autonomy with semantically-annotated hierarchical cases.

In *Proceedings of the 23rd International Conference on Case-Based Reasoning*, 88–103.

Floyd, M. W.; Karneeb, J.; Moore, P.; and Aha, D. W. 2017. A goal reasoning agent for controlling UAVs in beyond-visual-range air combat. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 4714–4721. AAAI Press.

Geib, C.; Weerasinghe, J.; Matskevich, S.; Kantharaju, P.; Craenen, B.; and Petrick, R. P. 2016. Building helpful virtual agents using plan recognition and planning. In *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Gillespie, K.; Molineaux, M.; Floyd, M. W.; Vattam, S. S.; and Aha, D. W. 2015. Goal reasoning for an autonomous squad member. In *Goal Reasoning: Papers from the ACS 2015 Workshops*, 52–67.

Golpayegani, F., and Clarke, S. 2016. Goal-based multi-agent collaboration community formation: A conceptual model. In *Workshop on Goal Reasoning at IJCAI-2016*.

Hoffman, R.; Klein, G.; and Mueller, S. 2017. Initial concepts and literature review for DARPA-XAI. Technical report from task area 2 (psychological model of explanation) on DARPA contract DARPA-BAA-16-53. Technical report, Institute for Human and Machine Cognition, Pensacola, FL.

Jaidee, U.; Muñoz-Avila, H.; and Aha, D. W. 2013. Case-based goal-driven coordination of multiple learning agents. In *Proceedings of the 21st International Conference on Case-Based Reasoning*, 164–178.

Klein, G.; Woods, D. D.; Bradshaw, J. M.; Hoffman, R. R.; and Feltovich, P. J. 2004. Ten challenges for making automation a ”team player” in joint human-agent activity. *IEEE Intelligent Systems* 19(6):91–95.

Molineaux, M.; Klenk, M.; and Aha, D. 2010. Goal-driven autonomy in a Navy strategy simulation. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence*, 1548–1554.

Richards, D., and Stedmon, A. 2017. Designing for human–agent collectives: display considerations. *Cognition, Technology & Work* 19(2-3):251–261.

Roberts, M.; Vattam, S.; Alford, R.; Auslander, B.; Karneeb, J.; Molineaux, M.; Apker, T.; Wilson, M.; McMahon, J.; and Aha, D. W. 2014. Iterative goal refinement for robotics. In *Planning and Robotics: Papers from the ICAPS Workshop*. AAAI Press.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*. MIT Press Cambridge.

Weber, B. G.; Mateas, M.; and Jhala, A. 2010. Applying goal-driven autonomy to starcraft. In *Proceedings of the Sixth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*.

Interaction and Learning in a Humanoid Robot Magic Performance

Kyle Morris, John Anderson, Meng Cheng Lau, Jacky Baltes
University of Manitoba, Winnipeg, MB R3T2N2, Canada

Abstract

Magicians have been a source of entertainment for many centuries, with the ability to play on human bias, and perception to create an entertaining experience. There has been rapid growth in robotics throughout industrial applications; where primary challenges include improving human-robot interaction, and robotic perception. Despite preliminary work in expressive AI, which aims to use AI for entertainment; there has not been direct application of fully embodied autonomous agents (vision, speech, learning, planning) to entertainment domains. This paper describes preliminary work towards the use of magic tricks as a method for developing fully-embodied autonomous agents. A card trick is developed requiring vision, communication, interaction, and learning capabilities all of which are coordinated using our script representation. Our work is evaluated quantitatively through experimentation, and qualitatively through acquiring 2nd place at the 2016 IROS Humanoid Application Challenge. A video of the live performance can be found at https://youtu.be/OMpcmcPWAVM.

Introduction

Humans have long enjoyed the clever trickery that comes from a good magic show. Magic tricks embody the primary features desired for an intelligent agent. These include **reactivity**: the ability to quickly perceive and respond to changes in the environment; **proactivity**: being goal-driven and acting towards reaching some desired goal; and **social ability**: the ability to communicate with others to further reach their goal (Wooldridge 2009).

Non-deterministic and dynamic environments pose challenges in developing robust autonomous agents that possess these features. This difficulty lies in balancing the proactive and reactive behaviour (Wooldridge 2009). An agent that is purely reactive may fail to reach a desired goal, whereas a purely proactive (goal driven) agent may not spend enough time acting to reach a goal (Wooldridge 2009).

During a live performance, reactivity is desired to provide authentic response time for each event in the script. Proactiveness involves seeking an end-performance goal that logically entails from the events in the script. The script is central to both reactivity and proactiveness. Lastly, social ability is required to leverage off the audience and guide a performance to cater towards their demographic and play off of their bias. For example, non-explicit humorous remarks are prioritized for an audience containing youth. Our work presents an autonomous agent that performs a magic card trick. We created motion, speech, and vision components on top of our custom DARwIn OP2 framework. These components utilize PocketSphinx for speech recognition, and OpenCV2 for playing card classification. The use of a finite state machine gives structure to the performance and allows the agent to seek an end-performance goal that accounts for potential problems that may arise during the show. Lastly, an easily adjustable design of events allows for a unique performance and user experience.

Related Work

Live performance takes many forms. Humanoid robotics competitions have explored the development of robust, versatile agents that perform multiple distinct sporting events autonomously (Baltes et al. 2016). Furthermore, teams of robots are used to research how cooperation techniques are used in reaching a desired goal (Ashar et al. 2015). Such competitions have grown in popularity and have evolved to use more entertaining events that remain as useful benchmarks (Gerndt et al. 2015), but do not yet cater easily to a non-research audience.

Expressive AI has explored artificial intelligence for pure entertainment purposes in domains that include games (Mateas 2003) and music (De Mántaras and Arcos 2002); but lacks a robotics implementation. In the domain of Robotics, work has been done on incorporating entertainment (Kuroki 2001) with further specialization into card magic (Koretake, Kaneko, and Higashimori 2015). This work however puts focus on card manipulation, and mechanical aspects rather than timing and interaction. There has been growing discussion of the need for timing and human-robot interaction for effective live performance (Nuñez et al. 2014; Tamura, Yano, and Osumi 2014); but this discussion has been purely theoretical. Our work outlines a new application of robot entertainment for live magic that incorporates computer vision, machine learning, speech recognition and motion in order to deliver an authentic and robust perfor-

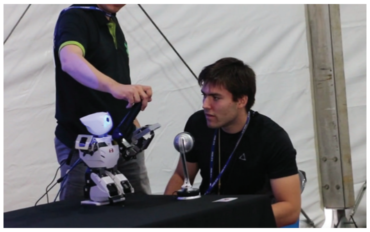


Figure 1: The live performance at IROS 2016. The robot is about to reveal the cards.

mance.

Employing template-matching for playing card recognition has demonstrated higher overall classification accuracy, but only in settings where the card is viewed from a fixed distance and angle (Brinks and White 2007). Similarly, this approach had significant latency (6 seconds) using a client-server architecture and has not yet been tested on a localized model (Brinks and White 2007). Work from (Zheng and Green 2007) demonstrated higher rank classification accuracy along with robustness to card rotation and scale, however there is no evaluation of the overall classification accuracy. Furthermore we achieved higher accuracy on Jack, Queen, and King cards, along with higher suit accuracy. Other approaches such as (Martins, Reis, and Teófilo 2011) achieved higher rank classification; but share similar challenges in suit classification. Despite marginally lower performance on rank classification, our system demonstrates significant overall classification accuracy while being robust to card rotation, translation, and scale.

The Magic Trick

The trick is based on the classic straight-man act, in which a stern robot assistant contrasts with a charismatic but condescending human magician. A DARwIn-OP2 robot is asked to select and observe 3 cards from a deck. Vocal cues from the human magician provoke responses from the robot. Throughout the performance the robot grows impatient with the magicians' rude gestures and treatment, and takes over the magic performance by knocking the deck out of the magicians' hand. After the robot acquires the deck, the robot explains the simplicity of the magic trick, and reveals the 3 cards that were originally chosen, from the face-down deck.

Problem Representation

We represent a performance as a collection of ordered phases. A phase is some discrete set of events that must take place together within a limited time. For example one phase may involve multiple listen-response events where an agent uses speech recognition and speech synthesis to follow dialog with a human magician. Another phase may rely on both motion gestures to hold a deck of cards, and computer vision to recognize playing cards.

Grouping events into phases allows for a graceful recovery from potential interrupts in the performance. If, for example, a dialog-only phase is taking place, and noise interference occurs, the agent may transfer to a backup phase which involves asking where the noise is coming from. During a card recognition phase that uses only the vision and motion components, it would not make sense for the agent to stop reading cards, or freeze up; because of the noise. It would make sense to have a backup phase in case the lighting is poor, in which the agent may ask for better lighting. The use of a state machine guides the performance by transitioning through pre-designed phases which together form a coherent story.

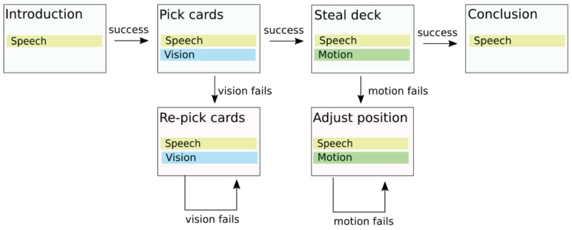


Figure 2: Phases of the performance

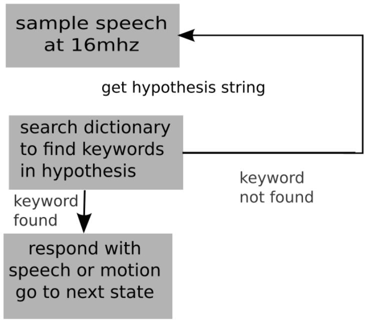

sample speech at 16mhz
get hypothesis string
search dictionary to find keywords in hypothesis
keyword not found
keyword found
respond with speech or motion go to next state

Figure 3: Control flow of speech processing

Implementation

Speech Recognition and Synthesis

Voice audio was recorded using a NESSIE Adaptive USB Condenser Microphone at 16kHz. Incoming audio is processed using PocketSphinx in order to generate a hypothesis string. This hypothesis string is checked against a custom language dictionary containing 89 keywords from the magic show script. If selected keywords are found in said string, this will trigger a response from the robot. Each dialog event may be customized to require multiple distinct keywords.

Vision

Input images are captured using the built-in DARwIn-OP2 Logitech camera and passed to a custom vision module. The vision module was built with C++ and OpenCV2. The input image is first preprocessed by gray scaling, applying blur, and then applying a binary threshold. Contours are then extracted from the image and organized into a hierarchical tree

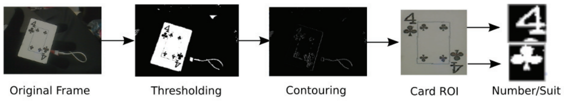


Figure 4: The vision pipeline

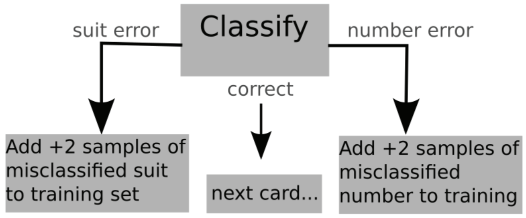

Classify
suit error
number error
correct
Add +2 samples of misclassified suit to training set
next card...
Add +2 samples of misclassified number to training

Figure 5: The training process

and compressed with OpenCV's simple chain approximation to gather only end-points of the contours. Polygon approximation is used on the contour to gather estimated corner points for a playing card. In order to eliminate false detection, the points are checked to be rectangular(based on the ratio between them). An affine transformation is used on the card ROI. Due to symmetry of playing cards, the bottom left corner is checked for a card symbol. If this symbol is missing, the card is assumed to be mirrored, and will be reflected to the correct orientation.

Card Classification Card suit (Diamonds, Hearts, Spades, Clubs) and rank (1-10, Jack, King, Queen, Ace) ROI are extracted. These ROI are then either dilated or eroded according to lighting in the environment. The suit and rank ROI are then classified using the K-Nearest Neighbours algorithm (Cover and Hart 1967).

Machine Learning

The training process took place using a deck of 52 cards. The initial training set contained $5_{images} \times 4_{suits} \times 13_{cards} = 260$ samples collected using the robots built-in camera. Each sample is stored as a 30x30 gray-scale image in csv format as a 1×900 matrix of pixel brightness values [0-255]. The K-Nearest Neighbours algorithm (Cover and Hart 1967) is used to classify each suit and rank. An iterative training process is used. Initially each card within the full deck is shown in front of the robot. If the card is correctly classified, it is placed in a success pile. Misclassification may take place on either the card rank or suit. In either case, the misclassification is recorded and 2 positive samples of this rank or suit are added to the training set. The card will then be placed in a fail pile. For example if a Two of Hearts is misclassified as a Two of Diamonds, we will add 2 positive samples of the Hearts suit to the training set. The next iteration will begin using cards from the fail pile. This iterative process terminates when the fail pile is empty.

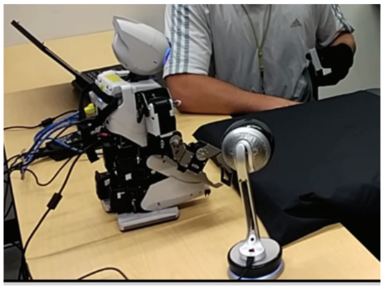


Figure 6: The dynamic evaluation setup.

Evaluation

Our iterative training process was used, yielding the final training set. The test set was then created by randomly shuffling the deck and placing each card in front of the robot. This process was repeated 5 times to create a total of $5_{samples} \times 13_{ranks} = 65$ test samples for each suit, and $5_{samples} \times 4_{suits} = 20$ test samples for each rank. Evaluation was first completed in a dynamic setting. This included exposure to daylight, and randomization from a human holding the card in front of the robot. A second controlled evaluation consisted of static lighting, and a fixed placement of each card on a black surface.

A rank classification accuracy of 89.23% across the 13 card ranks was achieved using the dynamic setting. This surpassed the controlled setting which achieved 83.46% accuracy. Similarly the dynamic setting achieved a higher classification accuracy (90.38%) than the controlled setting (83.46%) on card suits. It is interesting to note the difference in spread between the two evaluations. The controlled setting has a higher standard deviation (10.76% for card rank, 15.99% for card suit) than the dynamic setting (4.07% for card rank, 11.15% for card suit). We believe this is due to our system being trained in a more dynamic setting.

Conclusions and Future Work

This work explored the use of live entertainment in agent-based research. Specifically live magic performance was chosen as an avenue for developing a fully-embodied autonomous agent. Our card trick incorporates on-board vision, communication, interaction, and learning capabilities that allow for robust performance. This work may be greatly enhanced with improvements to the vision and machine learning components. Overall classification accuracy is dependent on both rank and suit accuracy. Our method demonstrated robustness to card rotation, translation and scale; but fell short in overall accuracy. We share similar challenges to other aforementioned vision techniques (Brinks and White 2007; Zheng and Green 2007; Martins, Reis, and Teófilo 2011), and believe improvements to image resolution would combat these challenges. Similarly, we see the use of colour recognition as a simple and promising approach to improve suit classification accuracy (Martins, Reis, and Teófilo 2011). Such improvements are challenging to acquire under time and space constraints imposed by on-board hardware. Lastly, we are interested in generalizing our work

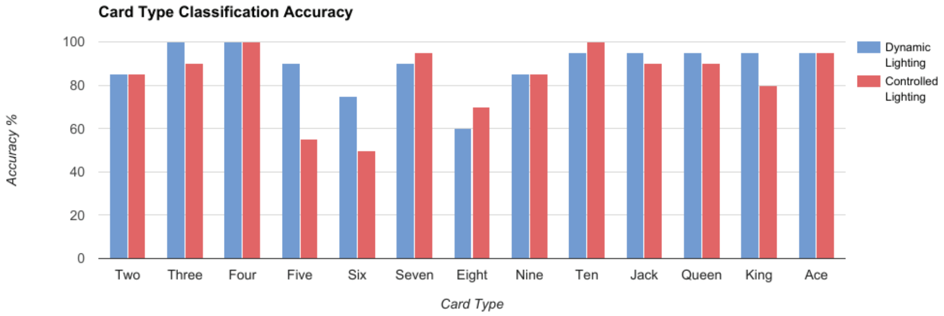


Figure 7: Classification results for card ranks. Taken from 20 samples of each card rank.

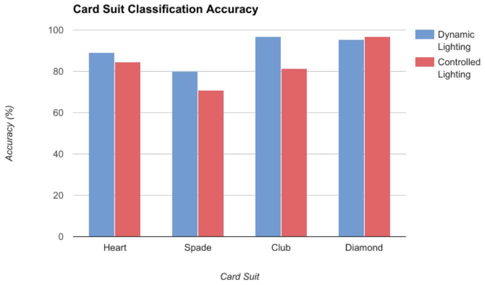


Figure 8: Classification results for card suits. Taken from 60 samples of each card suit.

into a framework for building agents capable of live performance. We believe this framework would provide easier entry, and thus encourage agent-based research using live entertainment.

References

Ashar, J.; Ashmore, J.; Hall, B.; Harris, S.; Hengst, B.; Liu, R.; Mei (Jacky), Z.; Pagnucco, M.; Roy, R.; Sammut, C.; Sushkov, O.; Teh, B.; and Tsekouras, L. 2015. *RoboCup SPL 2014 Champion Team Paper*. Cham: Springer International Publishing. 70–81.

Baltes, J.; Tu, K.-Y.; Sadeghnejad, S.; and Anderson, J. 2016. Hurocup: competition for multi-event humanoid robot athletes. *The Knowledge Engineering Review* 1–14.

Brinks, D., and White, H. 2007. Texas hold'em hand recognition and analysis.

Cover, T., and Hart, P. 1967. Nearest neighbor pattern classification. *IEEE transactions on information theory* 13(1):21–27.

De Mántaras, R. L., and Arcos, J. L. 2002. Ai and music: From composition to expressive performance. *AI Magazine* 23:43–58.

Gerndt, R.; Seifert, D.; Baltes, J. H.; Sadeghnejad, S.; and Behnke, S. 2015. Humanoid robots in soccer: Robots versus humans in robocup 2050. *IEEE Robotics & Automation Magazine* 22(3):147–154.

Koretake, R.; Kaneko, M.; and Higashimori, M. 2015. The robot that can achieve card magic. *ROBOMECH Journal* 2(1):5.

Kuroki, Y. 2001. A small biped entertainment robot. In *MHS2001. Proceedings of 2001 International Symposium on Micromechatronics and Human Science (Cat. No.01TH8583)*, 3–4.

Martins, P.; Reis, L. P.; and Teófilo, L. 2011. Poker vision: playing cards and chips identification based on image processing. In *Iberian Conference on Pattern Recognition and Image Analysis*, 436–443. Springer.

Mateas, M. 2003. Expressive ai: Games and artificial intelligence. In *DiGRA '03 - Proceedings of the 2003 DiGRA International Conference: Level Up*.

Nuñez, D.; Tempest, M.; Viola, E.; and Breazeal, C. 2014. An initial discussion of timing considerations raised during development of a magician-robot interaction. In *Proc. ACM/IEEE Workshop on Timing in Human-Robot Interaction HRI*.

Tamura, Y.; Yano, S.; and Osumi, H. 2014. Modeling of human attention based on analysis of magic. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, HRI '14, 302–303. New York, NY, USA: ACM.

Wooldridge, M. 2009. *An introduction to multiagent systems*. John Wiley & Sons.

Zheng, C., and Green, R. 2007. Playing card recognition using rotational invariant template matching. In *Proc. of Image and Vision Computing New Zealand*, 276–281.

Position Paper: Reasoning About Domains with PDDL

Alexander Shleyfman, Erez Karpas
Faculty of Industrial Engineering and Management
Technion — Israel Institute of Technology

Abstract

One of the major drivers for the progress in scalability of automated planners has been the introduction of the Planning Domain Definition Language (PDDL) and the International Planning Competition (IPC). While PDDL provides a convenient formalism to describe planning problems, there is a significant gap with regards to describing domains. Although PDDL is split into a domain description and a problem description, the domain description is not enough to specify a domain completely, as it does not constrain the possible problems in the domain. For example, there is nothing in the BLOCKSWORLD PDDL domain description which says that a block can not be on top of itself in the initial state. In this position paper, we argue that PDDL domains should be extended to incorporate a new section which constrains possible problems in the domain. We argue that such an extension can be based on first-order logic, and describe several use cases where this extension might be of use. We also provide some preliminary empirical results of one way for automatically extracting such constraints based on mutual exclusion.

Introduction

The domain-independent planning community has made significant progress scaling up planners, allowing them to address bigger and more complicated problem instances. One of the major drivers for this progress has been the introduction of the International Planning Competition (IPC), with its standard language for describing planning problems — PDDL, the Planning Domain Definition Language (McDermott 2000). The PDDL language was further extended to support additional features, which were introduced in later iterations of the IPC (Fox and Long 2003; Edelkamp and Hoffmann 2004; Gerevini and Long 2005).

PDDL splits the definition of a planning problem into two parts: domain and problem. The domain describes the types of objects this domain deals with, along with schemata for the predicates used to describe the state of the world and the operators used to change it. The problem describes the specific objects in the world in this problem instance, as well as the initial state and the goal. Typically, a domain in the IPC is defined by a single PDDL domain description (usually in a separate file), and a random problem instance generator[1]. Planners are then evaluated based on their performance on a set of problem instances generated by the problem generator.

While this is a reasonable way to evaluate how well planners solve planning *problems*, we claim that it is extremely difficult to reason over a *domain*. For example, consider the well known BLOCKSWORLD domain, which features the predicate ON(x, y), indicating that block x is directly on top of block y. We would like to be able to prove that a block can never be on top of itself. This is fairly easy to do using techniques such as relaxed reachability. However, relaxed reachability takes an initial state as input, and the initial state is only described in the PDDL problem. In fact, there is nothing preventing us from generating an instance of BLOCKSWORLD in which ON(A, A) does appear in the initial state. Thus, in order to be able to prove that a block is never on top of itself, we would need some *explicit* description of the fact that a block is never on top of itself in the initial state of any valid problem instance. Currently, this knowledge is only *implicit* from our understanding of the meaning of the domain.

Previous work (Helmert 2003) has defined a domain as an infinite set of grounded planning problems. While this definition is good enough to theoretically analyze the complexity of planning in a domain, it does not consider the issue of representation. In this position paper, we argue that the PDDL language needs to be further extended, in order to allow for automated reasoning about *domains*, rather than only single problem instances. We argue that such an extension can be based on first-order logic, and describe several use cases where this extension might be of use. We also provide some preliminary empirical results where we can identify what are *probably* domain-level mutual exclusion (mutex) groups, providing some automated support for encoding our suggested constraints.

Background

We begin with a brief review of PDDL. For the full details, we refer the reader to the various papers describing the different versions of PDDL (McDermott 2000; Fox and Long 2003; Edelkamp and Hoffmann 2004; Gerevini and Long

[1] Some domains have a separate domain description for each problem instance. We will address this issue in the final discussion.

2005). As previously mentioned, PDDL divides the definition of a planning problem into two parts: the domain, and the problem, which typically are contained in two different files. The division allows for the same domain file to be used with multiple problem files.

A PDDL domain consists of a description of the possible types of objects in the world. A type t can inherit from another type s, so that all objects with type s are also of type t. While there is a small controversy regarding whether the type hierarchy must form a proper tree, or can be a graph, this issue is irrelevant for the purposes of this paper. The domain also consists of a set of constants, which are objects which appear in all problem instances of this domain.

The second part of the domain description is a set of predicates. Each predicate is described by a name and a signature, consisting of an ordered list of types. Given a set of objects, we can *ground* the given predicates, yielding a set of propositions which describe the state of the world. Note, however, that these objects are only given as part of the problem description, and not in the domain description. The domain also describes a set of derived predicates, which are predicates associated with a logical expression. The idea is that the value of each derived predicate is computed automatically by evaluating the logical expression associated with it.

Finally, the domain description consists of a set of operators. Each operator is also described by a name, a signature, a precondition, and an effect. The signature is now an ordered list of named parameters, each with a type. The precondition is a logical formula, whose basic building blocks are the above mentioned predicates, combined using the standard first order logic logical connectives. We remark that the predicates can only be parametrized by the operator parameters, the domain constraints, or, if they appear within the scope of a forall or exists statement, by the variable introduced by the quantifier. The effect of the operator is similar, except that it described a partial assignment, rather then a formula, and thus can not contain any disjunctions. An operator can also be grounded given a set of objects, yielding grounded actions.

A PDDL problem is much simpler than the domain. It consists of a set of objects, each associated with a type (if a type is not specified, the object is assumed to be of a default type), and a description of the initial state and the goal. The initial state is described by the list of propositions (grounded predicates) that are true in it, where any proposition that is not listed it assumed to be false. The goal is also a logical expression, similarly to the precondition of an operator, except that it can refer to all objects in the problem instance. Although the goal can be an arbitrarily complex logical expression, in most existing planning benchmarks domains, it is a simply conjunction of positive propositions. In the rest of this paper, we will assume the goal takes this simple form, and discuss more complex goals in the conclusion.

As mentioned above, a domain can be grounded given a set of objects, which are described in the problem. Most modern planners start by grounding the given planning problem, and operate on the grounded problem description. However, if our intention is to reason over a domain, this approach is not practical, as there is no single problem to ground over. In the next section, we present our proposal to extend PDDL to allow some reasoning over a domain, even when a problem instance is not given.

```
(forall (?x) (not (init (on ?x ?x))))

(forall (?x ?y ?z) (implies
(and (init (on ?y ?x)) (init (on ?z ?x)))
(= ?z ?y)))

(forall (?x) (or
(init (on-table ?x))
(exists (?y) (init (on ?x ?y)))))

(forall (?x) (not (goal (on ?x ?x))))

(forall (?x ?y ?z) (implies
(and (goal (on ?y ?x)) (goal (on ?z ?x)))
(= ?z ?y)))
```

Figure 1: BLOCKSWORLD Domain Constraints

Extending PDDL

The heart of our proposed extension to PDDL is to add *constraints* about the problem to the domain description. Following the BLOCKSWORLD example from the introduction, we could specify that in the initial state of any legal instance of BLOCKSWORLD, no block is on top of itself.[2] We could then use a lifted version of relaxed reachability analysis to infer that no block can ever be on top of itself.

Specifically, we propose to add another section to the PDDL domain description, which will consist of a set of constraints. Each constraint will be a first order logic statement, which can refer to domain constants, and, of course, to variables introduced by each quantifier within its scope. However, the basic building blocks will not be predicates, but rather predicates perpended with a modal operator, specifying if this refers to the initial state or the goal. One caveat is that we can not check whether some proposition if false in the goal, as the goal is only a partial state. We also explicitly allow the usage of the (object) equality predicate. As the following examples will show, it is quite useful.

The interpretation of these constraints is, naturally, as constraints over a problem description. We can treat each problem as specifying a full initial state, and a partial goal state (as we assume the goal only describes the propositions we want to be true). Thus, we can evaluate each constraint, and check whether a given problem satisfies it.

Figure 1 shows how our extension can be applied to BLOCKSWORLD. The first constraint states that a block is never on top of itself in the initial state. The second constraint states that there can be at most one block on top of another block (i.e., if y and z are both on top of x, then they must be the same block). The third constraint states that every block must be on top of another block or on the table in

[2] These are different than the constraints introduced in PDDL 3.0 (Gerevini and Long 2005), which constrain possible *plans* for a given problem.

the initial state. Finally, the last two constraints are similar to the first two, except they are applied to the goal.

Another example highlights the differences between two different versions of the LOGISTICS domain: the one used in the first IPC (1998) and the one used in the second IPC (2000). Even though the PDDL domain description was the same in both competitions, LOGISTICS-98 is still much harder to solve than LOGISTICS-2000. This is because in the instances generated for second IPC, there was an implicit constraint, that there is exactly one truck in each city. This constraint is shown in Figure 2.

Use Cases

So far, we have only proposed an extension to PDDL, without explaining why we believe such an extension is useful. In this section we provide several use cases where our proposed extension can be useful. We remark that we have not implemented any of these ideas, we simply claim that these can be the subject of future work.

Learning and Using Domain Control Knowledge

There has been a significant body of work on learning, and using, domain control knowledge. While a full review of all the relevant literature is beyond the scope of this paper, we review some influential works in this area. First, the original STRIPS system had a macro learning component, which attempted to generalize successful plans from one problem to others (Fikes, Hart, and Nilsson 1972). This is, in fact, an example of explanation based learning (EBL) (e.g., (Mooney and Bennett 1986; Minton 1990)), where a system typically look at a single example and attempts to generalize it.

Another example is the TLPlan planner (Bacchus and Kabanza 2000), which was able to exploit manually coded domain-specific control knowledge expressed in a temporal logic. Later work tried to learn such rules automatically (Yoon, Fern, and Givan 2008). In fact, the learning track in the international planning competition (IPC), introduced in 2011 (Fern, Khardon, and Tadepalli 2011), focuses on learning domain control knowledge. In the learning track, each competitor is given access to a PDDL domain file and a random problem generator. The competitor is then given a very long time to produce a domain control knowledge (DCK) file, which the planner can then use to solve new problems in the domain, with the intent that the DCK will help the planner improve its performance.

With the way this track is set up, the best type of guarantee that can be provided is a probably approximately correct (PAC) (Valiant 1984) style guarantee, i.e., that there is a high probability that the learner has learned something that is fairly good. However, there is no way to guarantee that the learned domain control knowledge will work, because there is no characterization of all possible instances in the domain, but only a sample of problem instances. Adopting the proposed extension to PDDL will allow learners to *prove* something about what they are learning.

For example, suppose we wanted to make the Fast Downward translator (Helmert 2009) more efficient by learning what propositions are grouped together into a finite-domain variable. We might be able to learn, for example, that the location of a truck in LOGISTICS is always a mutex group, and can thus be used to create a finite domain variable. In fact, since the translator looks for invariants in a lifted way in the domain and then generates possible mutex groups from invariants which have a single matching fact that is true in the initial state, it is relatively straightforward to so, as our preliminary empirical results demonstrate. Of course, this is only possible if we know that AT(T, L) has exactly one true proposition for each given truck T in the initial state — something which is easily described using our proposed PDDL extension.

Similar invariants can be seen in the BLOCKSWORLD domain. The same as trucks, blocks each are represented as single finite domain variables, which are generated using the invariants founded in the domain description, and the predicates in the initial state. These mutexes however, are not enough to randomly generate a “realistic” BLOCKSWORLD problem. As we mentioned before, a single block can not be placed on itself, and thus there are no predicates of the form ON(x, x) in the initial state. However, consider a problem with two blocks A and B, where block A is placed on top of block B and block B is placed on top of block A. It is easy to see that this position satisfies the condition described in the previous section, but in the same time, it’s both “unrealistic” and unsolvable, given the blocks A and B have some other positions in the goal description. Even more so, this “ouroboros”[3] of a sort can be extended to a cycle of an arbitrary length, making this condition hard to detect without some recursive logical formula. Thus, our proposed extension must be able to support recursive formulas, to be able to express these restrictions.

Generalized Planning

A somewhat similar use case occurs in generalized planning. In generalized planning, the objective is to generate a controller which can solve all possible problems from a given planning domain. Examples of work on generalized planning include generating plans with loops and branching (Srivastava, Immerman, and Zilberstein 2011) and finite state controllers (Bonet, Palacios, and Geffner 2009; Aguas, Celorrio, and Jonsson 2016). Again, the issue is that with no formal specification of a domain, it is impossible to prove that a controller will solve all problems in a domain.

On the other hand, using our proposed PDDL extension, it is very easy (in theory) to use the following scheme. First, call a generalized planner on a given set of problems in the domain of interest. Second, verify if the resulting controller solves all possible problems in the domain. If the answer is yes, we have a controller that can solve all problems in the domain. Otherwise, generate a counter example, add it to the given set of problems, and repeat. Of course, the problem of verifying if the given controller works for all possible problems in the domain, and generating a counter example if it does not is undecidable (as we can generate a domain that corresponds to a Turing machine, and each problem corre-

[3] A serpent eating its own tail.

```
(forall (?c - city ?s ?t - truck) (implies
   (and (exists (?l - location) (and (init (in-city ?l ?c)) (init (in ?t ?l))))
        (exists (?l - location) (and (init (in-city ?l ?c)) (init (in ?s ?l)))))
   (= ?s ?t)))
```

Figure 2: LOGISTICS-2000 Additional Constraint

sponds to a given instance terminating). Nevertheless, efficient (incomplete) termination analyzers do exist, thus allowing us to hope this idea might work in practice on some domains of interest.

Almost Automatic Random Problem Generators

When creating a new domain in PDDL, the burden of specifying which problems are legal and which are not falls to the problem generator. For example, the problem generator for BLOCKSWORLD will never generate a problem in which on(A, A) appears in the initial state. However, this knowledge is part of the problem generator's code. On top of this, the problem generator provides some distribution on the problems.

With our proposed extension, the first part of the random problem generator's job could be automated. The only implementation necessary in a random problem generator would be just the random part — the distribution.

While we believe this would be beneficial by itself, this also has the potential of enabling bootstrapping approaches (Arfaee, Zilles, and Holte 2010), where larger and larger problem instances must be generated. Of course, the issue of where the distribution comes from is still a critical component of such an approach, which is beyond the scope of our proposed PDDL extension.

State Estimation

Finally, another use case comes from the combination of planning with real world sensing. Consider, for example, a camera looking at a BLOCKSWORLD scene. The camera, along with the image processing and object recognition software that looks at its output, will typically produce a set of real-world coordinates for the position of each block. These coordinates will typically have some error associated with them, due to sensor noise, lighting conditions, probabilistic image processing algorithms, and more.

A state estimator will look at the history of these measurements to produce the symbolic description of the current state. Without telling the state estimator that a block can only be on top of one other block, we might end up with states containing both on(A, B) and on(A, C). However, if our state estimator was able to infer mutual exclusion invariants for the domain, it could reject samples which violate these constraints, yielding more accurate state estimates.

Case Study: Discovering Domain Mutexes

As a first step to demonstrate reasoning over a domain, rather than over individual problems, we used the Fast Downward translator (Helmert 2009), in order to examine invariant candidates in PDDL domains from IPC benchmarks. As previously mentioned, the Fast Downward translator identifies lifted invariant candidates looking only at the PDDL domain. For example, the translator identifies that for a given truck T, the number of locations L for which AT(T, L) holds does not increase for any applicable action. The translator then checks whether this invariant candidate generates a set of mutexes, by checking if the number of locations each truck T is at in the initial state is 1 or less.

In this case study, we used the invariants discovered by the Fast Downward translator for each domain. For each invariant, we checked whether it always led to mutexes in all instantiations of the invariant in all problems. If so, then it is likely safe to add a problem constraint derived from this invariant to the domain. However, without an explicit extension to PDDL, we can never know that this is a true lifted mutex, or whether the random problem generator just happened to only generate problems where this invariant happened to lead to mutexes.

Experimental Results

For our experiment we used the International Planning Competition benchmarks (IPC'98 – IPC'11), from which we excluded all the benchmarks that have more than one domain description file. In the relevant benchmarks we count the invariant candidates extracted by the Fast Downward translator, and check which of those invariant lead to mutex groups, and which did not (due to the fact that the number of initial state propositions participating in these invariants exceeded 1). The results are presented in Table 1. Note that there are no domain invariants that have not been grounded to a mutex group due to the absence of the appropriate initial states facts.

Most of the invariants in these benchmarks are either always mutex groups, or always *overcrowded* – there are at least 2 propositions in the initial state that participate in that invariant. However, there are some invariants that are mixed, that is, lead to mutex groups in some cases, and are overcrowded in others. Detailed analysis shows that this happens mostly due to the fact that there is a smaller invariant that is contained in a larger one. For example, in the LOGISTICS domain all the locations of a given truck T constitute an invariant, but all the locations of all the trucks are also an invariant of the domain. The latter invariant leads to a mutex group only in the case where there is exactly one truck in the problem. Thus, mixed invariants can be seen grounded in the small problems of the domains, but getting overcrowded in the large ones.

Conclusion

In this paper, we have proposed an extension to PDDL which will allow for automated formal reasoning about domains. This extension will make no difference to the task of solving a single planning problem, with the possible exception of first validating the given problem instance. However, as we

Domain	Inv	Pure	Over	Mixed
AIRPORT-ADL	8	6	0	2
ASSEMBLY	0	0	0	0
BARMAN-OPT11-STRIPS	3	3	0	0
BARMAN-SAT11-STRIPS	3	3	0	0
BLOCKS	3	3	0	0
DEPOT	5	4	1	0
DRIVERLOG	2	2	0	0
ELEVATORS-OPT11-STRIPS	3	3	0	0
ELEVATORS-SAT11-STRIPS	3	3	0	0
FLOORTILE-OPT11-STRIPS	5	4	1	0
FLOORTILE-SAT11-STRIPS	5	4	1	0
FREECELL	7	6	1	0
GRID	7	5	2	0
GRIPPER	3	3	0	0
LOGISTICS00	1	1	0	0
LOGISTICS98	1	1	0	0
MICONIC-SIMPLEADL	1	1	0	0
MICONIC	1	1	0	0
MOVIE	0	0	0	0
MPRIME	3	3	0	0
MYSTERY	3	3	0	0
NO-MPRIME	2	2	0	0
NO-MYSTERY	3	3	0	0
NOMYSTERY-OPT11-STRIPS	2	2	0	0
NOMYSTERY-SAT11-STRIPS	2	2	0	0
OPENSTACKS	8	5	3	0
OPTICAL-TELEGRAPHS	7	6	1	0
PARKING-OPT11-STRIPS	4	3	1	0
PARKING-SAT11-STRIPS	4	3	1	0
PEGSOL-OPT11-STRIPS	2	1	1	0
PEGSOL-SAT11-STRIPS	2	1	1	0
PHILOSOPHERS	7	6	1	0
PIPESWORLD-NOTANKAGE	2	1	1	0
PSR-LARGE	0	0	0	0
PSR-MIDDLE	0	0	0	0
ROVERS	12	6	3	3
SATELLITE	2	1	0	1
SCANALYZER-OPT11-STRIPS	0	0	0	0
SCANALYZER-SAT11-STRIPS	0	0	0	0
SOKOBAN-OPT11-STRIPS	3	2	1	0
SOKOBAN-SAT11-STRIPS	3	2	1	0
STORAGE	3	3	0	0
TIDYBOT-OPT11-STRIPS	3	3	0	0
TIDYBOT-SAT11-STRIPS	3	3	0	0
TPP	5	5	0	0
TRANSPORT-OPT11-STRIPS	2	2	0	0
TRANSPORT-SAT11-STRIPS	2	2	0	0
TRUCKS	3	3	0	0
VISITALL-OPT11-STRIPS	1	1	0	0
VISITALL-SAT11-STRIPS	1	1	0	0
WOODWORKING-OPT11-STRIPS	7	6	1	0
WOODWORKING-SAT11-STRIPS	7	6	1	0
ZENOTRAVEL	2	2	0	0

Table 1: Inv – number of invariants in the domain; Pure – number of invariants that are always grounded; Over – number of invariants that always have at least two predicates in the initial state; Mixed – number of invariants that sometimes are grounded, and sometimes have to many predicted in the initial state.

have illustrated in the previous section, such an extension will allow us to perform formal reasoning over a domain description, as well as provide a cleaner definition of what constitutes a planning domain.

While the focus of this paper has been on classical planning, our proposal becomes perhaps even more relevant in the context of non-deterministic planning. Specifically, finite state controllers are very useful with non-deterministic and partially observable planning problems, and state estimation is a must for realistic applications that involve sensing in a partially observable world.

The paper does not presume to provide the definitive, best possible, extension to PDDL. Two issue that were already mentioned are that some domains have a separate domain description for each problem instance, and that the goal can be a logical formula, not just a single conjunction. With regards to the first issue, this is usually the result of simplifying ADL (Pednault 1989) to STRIPS for the sake of planners that can not handle ADL. We argue this is not a real issue here, as reasoning over our proposed extension will require ADL-like reasoning (specifically, quantifiers). Furthermore, it is possible to perform reasoning over a domain using the complex ADL domain specification, and then planning using the simplified STRIPS version of the *given problem*.

The second issue, of complex goals, deserves further discussion. It could be possible to modify our proposed extension to PDDL to contain more general statements about the goal, such as "the goal entails X" or "the goal contains X as a subexpression in a location specified by y". We are skeptical that such statements would be of use in modeling domains of interest to the planning community, and so we do not propose them here.

Finally, despite the issues mentioned above, we believe this paper serves as a starting point for a discussion about what exactly constitutes a domain, and on what the automated planning community can contribute on top of state-of-the-art automated planners.

References

Aguas, J. S.; Celorrio, S. J.; and Jonsson, A. 2016. Generalized planning with procedural domain control knowledge. In *Proc. IJCAI 2016*, 285–293.

Arfaee, S. J.; Zilles, S.; and Holte, R. C. 2010. Bootstrap learning of heuristic functions. In *Proc. SoCS 2010*, 52–60.

Bacchus, F., and Kabanza, F. 2000. Using temporal logics to express search control knowledge for planning. *AIJ* 116(1-2):123–191.

Bonet, B.; Palacios, H.; and Geffner, H. 2009. Automatic derivation of memoryless policies and finite-state controllers using classical planners. In *Proc. ICAPS 2009*.

Edelkamp, S., and Hoffmann, J. 2004. PDDL2.2: The language for the classical part of the 4th International Planning Competition. Technical Report 195, Albert-Ludwigs-Universität Freiburg, Institut für Informatik.

Fern, A.; Khardon, R.; and Tadepalli, P. 2011. The first learning track of the international planning competition. *Machine Learning* 84(1-2):81–107.

Fikes, R. E.; Hart, P. E.; and Nilsson, N. J. 1972. Learning and executing generalized robot plans. *AIJ* 3:251–288.

Fox, M., and Long, D. 2003. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *JAIR* 20:61–124.

Gerevini, A., and Long, D. 2005. Plan constraints and preferences in PDDL3. Technical Report R. T. 2005-08-47, Dipartimento di Elettronica per l'Automazione, Università degli Studi di Brescia.

Helmert, M. 2003. Complexity results for standard benchmark domains in planning. *AIJ* 143(2):219–262.

Helmert, M. 2009. Concise finite-domain representations for PDDL planning tasks. *AIJ* 173:503–535.

McDermott, D. 2000. The 1998 AI Planning Systems competition. *AI Magazine* 21(2):35–55.

Minton, S. 1990. Quantitative results concerning the utility of explanation-based learning. *AIJ* 42(23):363–391.

Mooney, R. J., and Bennett, S. 1986. A domain independent explanation-based generalizer. In *Proc. AAAI 1986*, 551–555.

Pednault, E. P. D. 1989. ADL: Exploring the middle ground between STRIPS and the situation calculus. In *Proc. KR 1989*, 324–332.

Srivastava, S.; Immerman, N.; and Zilberstein, S. 2011. A new representation and associated algorithms for generalized planning. *AIJ* 175(2):615–647.

Valiant, L. G. 1984. A theory of the learnable. *CACM* 27(11):1134–1142.

Yoon, S.; Fern, A.; and Givan, R. 2008. Learning control knowledge for forward search planning. *JMLR* 9:683–718.

On Chatbots Exhibiting Goal-Directed Autonomy in Dynamic Environments

Biplav Srivastava
IBM Research

Abstract

Conversation interfaces (CIs), or chatbots, are a popular form of intelligent agents that engage humans in task-oriented or informal conversation. In this position paper and demonstration, we argue that chatbots working in dynamic environments, like with sensor data, can not only serve as a promising platform to research issues at the intersection of learning, reasoning, representation and execution for goal-directed autonomy; but also handle non-trivial business applications. We explore the underlying issues in the context of *Water Advisor*, a preliminary multi-modal conversation system that can access and explain water quality data.

Introduction

Chatbots (McTear, Callejas, and Griol 2016), which can engage people in natural dialog conversation, have gained popularity recently drawn by numerous platforms to create them quickly for any domain (Accenture 2016). Most common types of such agents deal with a single user at a time and conduct informal conversation, answer the user's questions or provide recommendations in a given domain. They need to handle uncertainties related to human behavior and natural language, while conducting dialogs to achieve system goals. Chatbots have been deployed in customer care in many industries where they are expected to save over $8 billion per annum by 2022 (Juniper 2017).

However, the data sources used by common chatbots are static databases like product catalogs or user manuals. Therefore, for their problem of dialog management, i.e., creating dialog responses to user's utterances, effective approaches include learning policies over predictable nature of data(Young et al. 2013) or reasoning on its abstract representations (Inouye 2004).

The application scenarios become more compelling when the chatbot works in a dynamic environment, e.g., with sensor data, and interacts with groups of people, who come and go, rather than only an individual at a time. In such situations, the agent has to execute actions to monitor the environment, model different users engaged in conversation over time and track their intents, learn patterns and represent them, reason about best course of action given goals and system state, and execute conversation or other multi-modal actions.

We now explore the underlying issues of goal-directed autonomy in dynamic environment in the context of *Water Advisor* (WA) (Ellis et al. 2018), a prototypical multi-modal conversation system that can access and explain water quality data to a variety of stake-holders. We identify opportunities for learning, reasoning, representation and execution in WA and motivate more such applications.

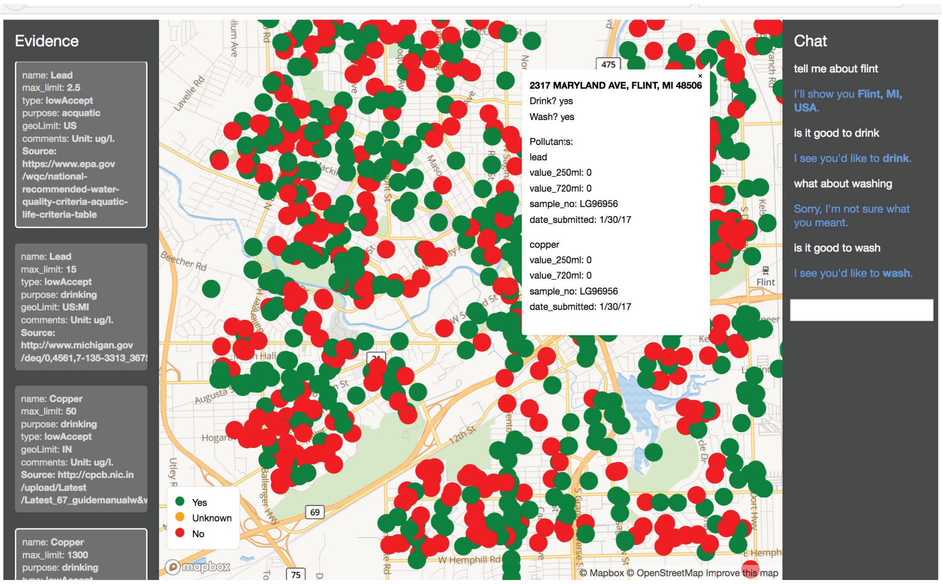
Decision-Support for Water Usage With a Multi-Modal Conversation Interface

The global situation of water quality around the world is alarming in both developing and developed countries((UNEP) 2016) because water demand continues to rise while existing sources for fresh water are getting polluted. A key strategy for tackling water pollution is engaging people. A person makes many daily decisions touching on water usage activities like for profession (e.g., fishing, irrigation, shipping), recreation (e.g, boating), wild life conservation (e.g., dolphins) or just regular living (e.g., drinking, bathing, washing). Accessible tools for public are particularly useful to handle public health challenges such as the Flint water crisis (Pieper, Tang, and Edwards 2017).

A decision in this space needs to consider the activity (purpose) of the water use; relevant water quality parameters and their applicable regulatory standards for safety; available measurement technology, process, skills and costs; and actual data. There are further complication factors: there may be overlapping regulations due to geography and administrative scope; one may have to account for alternative ways to measure a particular water quality parameter that evolves over time; and water data can have issues like missing values or at different levels of granularity. The very few tools available today target water experts such as WaterLive mobile app for Australia [1], Bath app for UK[2], and GangaWatch for India (Sandha, Srivastava, and Randhawa 2017) and assume technical understanding of sciences.

[1] http://www.water.nsw.gov.au/realtime-data

[2] 2https://environment.data.gov.uk/bwq/profiles/



Evidence

name: **Lead**
max_limit: **2.5**
type: **lowAccept**
purpose: **acquatic**
geoLimit: **US**
comments: **Unit: ug/l.**
Source: https://www.epa.gov /wqc/national-recommended-water-quality-criteria-aquatic-life-criteria-table

name: **Lead**
max_limit: **15**
type: **lowAccept**
purpose: **drinking**
geoLimit: **US:MI**
comments: **Unit: ug/l.**
Source: http://www.michigan.gov /deq/0,4561,7-135-3313_3675

name: **Copper**
max_limit: **50**
purpose: **drinking**
type: **lowAccept**
geoLimit: **IN**
comments: **Unit: ug/l.**
Source: http://cpcb.nic.in /upload/Latest /Latest_67_guidemanualw&v

name: **Copper**
max_limit: **1300**
purpose: **drinking**

2317 MARYLAND AVE, FLINT, MI 48506
Drink? yes
Wash? yes

Pollutants:
lead
value_250ml: 0
value_720ml: 0
sample_no: LG96956
date_submitted: 1/30/17

copper
value_250ml: 0
value_720ml: 0
sample_no: LG96956
date_submitted: 1/30/17

Chat

tell me about flint

I'll show you **Flint, MI, USA**.

is it good to drink

I see you'd like to **drink**.

what about washing

Sorry, I'm not sure what you meant.

is it good to wash

I see you'd like to **wash**.

Figure 1: A screenshot of *Water Advisor*. See video of it in action at https://youtu.be/z4x44sxC3zA.

Water Advisor

Water Advisor (WA) is intended to be a data-driven assistant that can guide people without requiring any special water expertise. One can trigger it via a conversation to get an overview of water condition at a location, explore it by filtering and zooming on a map, and seek details on demand (Figure 1) by exploring relevant regulations, data or other locations. The current prototype uses water quality data available from Flint, MI[3] but future extensions will use open water data from US Geological Survey[4] (USGS) that is refreshed for thousands of places in US per day. However, the number of water quality parameters, for which data is available, varies widely between locations and over time, making generation of useful advice challenging. For regulations, WA relies on information provided by multiple agencies at national (US, India) and state levels (Michigan, New York), which has been consolidated for reuse[5].

Technical Issues

In a water advising application, one or more users may need to interact with the chatbot if handling a complex decision like water contamination. The tool has to detect the user's information goals and meet them at lowest cognitive cost. The system uses a natural language classifier (NLC) to understand user utterance, and its error rate varies with input. The system has to decide whether to ask clarifying questions if it has low confidence and there are many ways to respond. The user may have preferences about how they specify an input (like location) and the kind of response they want (visual v/s textual). We discuss a range of issues below for exposition *but note that the current WA prototype handles only a subset of following integration issues.*

Learning plays an important role in understanding user's utterance, finding reliable water data samples in the database based on region and duration of interest, discovering issues in water quality and improving overall performance over time. In the prototype, for utterances, we use trained user models from commercial systems and for water quality, a simple regression method.

Representation is needed to map water's usage purpose to quality parameters and model safe limits of pollution parameters with different mathematic properties (e.g., polarity). It also helps map water purpose to regulations and further, aggregate and reconcile the latter when a region falls under overlapping jurisdiction of regulations. We represent this as geographically-scoped attribute-value pair in JSON format and make it publicly available for others to use and extend[6].

Reasoning is crucial to keep conversation focused based on system usability goals and user needs. One can model cognitive costs to user based on alternative system response choices and seek to optimize short-term and long-term behavior. Reasoning can further help to short-list regulations based on water activity and region of interest, generate advice and track explanations. We currently use rules on geographical scope and missing values to determine system response.

Execution is autonomous as the agent can choose to act by (a) asking clarifying questions about water usage goals or locations, (b) asking user's preference about advice, (c) seeking most reliable water data for region and time interval of interest from available external data sources, and corre-

[3] http://www.michigan.gov/flintwater/0,6092,7-345-76292_76294_76297—,00.html

[4] https://waterdata.usgs.gov/nwis/current/?type=quality

[5] https://github.com/biplav-s/water-info

[6] https://github.com/biplav-s/water-info

sponding subset of compatible regulations (d) invoking reasoning to generate an advice for water usage using filtered water data and regulations, (e) visualizing and explaining its output using water regulations, and (f) using one or more suitable modalities available at any turn of user interaction, i.e., chat, maps and document views. The current prototype uses a simplistic strategy for execution based on error rates, system confidence and usability rules.

Human Usability Factors have to be modeled and supported during WA's operation. In the current prototype, the user-interface controller module automatically keeps the different modalities synchronized so that the user is looking at consistent information across them. The system has to be aware of missing data or assumptions it is making, and needs to take them into account while communicating output advice in generated natural language. One avenue for future exploration is to measure and track complexity of interaction (Liao, Srivastava, and Kapanipathi 2017) and use sensed signals to pro-actively improve user experience. Another is to combine close-ended and open-ended questioning strategies for efficient interaction (Zhang, Liao, and Srivastava 2018).

Ethical Issues can emerge whenever a piece of technology is used among people at large. In the context of conversations, a recent paper surveys ethical issues (Henderson et al. 2018) like biases, adversarial examples, privacy violations, safety challenges and reproducibility concerns. A water-use chatbot can conceivably create bias among users of different activity subgroups (e.g., preferring recreation over drinking), compromise on privacy of users who submit queries about an activity or a region, and create public safety concerns (e.g., when users find scarcity of good quality water). We have not considered them in the prototype, however.

Discussion and Conclusion

In this paper, we used decision-support in water as a usecase to demonstrate that chatbots can serve as a promising platform to integrate AI sub-disciplines for goal-directed autonomy. Apart from learning, reasoning, representation and execution, chatbots also need to work with human usability factors and ethical issues. An interesting aspect of these applications is that the chatbot may be helping a group of people take collective decision making, like conducting an interview, and data changes over time. Beyond water and customer support, complex applications are emerging in sciences (astronomy(Kephart et al. 2018)), business (career counseling[7], hospitality[8]) and societal domains (health[9]).

References

Accenture. 2016. Chatbots in customer service. In *At: https://accntu.re/2z9s5fH*.

Ellis, J.; Srivastava, B.; Bellamy, R.; and Aaron, A. 2018. Water advisor - a data-driven, multi-modal, contextual assistant to help with water usage decisions. In *Proc. 32nd AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, Lousiana, USA.*

Henderson, P.; Sinha, K.; Angelard-Gontier, N.; Ke, N. R.; Fried, G.; Lowe, R.; and Pineau, J. 2018. Ethical challenges in data-driven dialogue systems. In *Proc. of AAAI/ACM Conference on AI Ethics and Society (AIES-18), New Orleans, Lousiana, USA.*

Inouye, R. B. 2004. Minimizing the length of non-mixed initiative dialogs. In Leonoor van der Beek, Dmitriy Genzel, D. M., ed., *ACL 2004: Student Research Workshop*, 7–12. Barcelona, Spain: Association for Computational Linguistics.

Juniper. 2017. Chatbots: Retail, ecommerce, banking & healthcare 2017-2022. In *At: http://bit.ly/2sJHejY*.

Kephart, J.; Dibia, V.; Ellis, J.; Srivastava, B.; Talamadupula, K.; and Dholakia, M. 2018. Cognitive assistant for visualizing and analyzing exoplanets. In *Proc. 32nd AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, Lousiana, USA.*

Liao, Q.; Srivastava, B.; and Kapanipathi, P. 2017. A Measure for Dialog Complexity and its Application in Streamlining Service Operations. *ArXiv e-prints*.

McTear, M.; Callejas, Z.; and Griol, D. 2016. Conversational interfaces: Past and present. In *The Conversational Interface. Springer, DOI: https://doi.org/10.1007/978-3-319-32967-3_4*.

Pieper, K. J.; Tang, M.; and Edwards, M. A. 2017. Flint water crisis caused by interrupted corrosion control: Investigating "ground zero" home. *Environmental Science & Technology* 51(4):2007–2014.

Sandha, S. S.; Srivastava, B.; and Randhawa, S. 2017. The gangawatch mobile app to enable usage of water data in every day decisions integrating historical and real-time sensing data. *CoRR* abs/1701.08212.

(UNEP), U. N. E. P. 2016. A snapshot of the worlds water quality: Towards a global assessment. In *Nairobi, Kenya. Online at: https://uneplive.unep.org/media/docs/assessments/unep_wwqa_report_web.pdf*.

Young, S.; Gašić, M.; Thomson, B.; and Williams, J. D. 2013. Pomdp-based statistical spoken dialog systems: A review. *Proceedings of the IEEE* 101(5):1160–1179.

Zhang, Y.; Liao, V.; and Srivastava, B. 2018. Towards an optimal dialog strategy for information retrieval using both open-ended and close-ended questions. In *Proc. Intelligent User Interfaces (IUI 2018, Tokyo, Japan, March.*

[7] https://www.ibm.com/talent-management/career-coach

[8] https://www.bebot.io/hotels

[9] https://www.healthtap.com/

Safe Goal-Directed Autonomy and the Need for Sound Abstractions

Siddharth Srivastava
School of Computing, Informatics and Decision Systems Engineering
Arizona State University
Tempe, AZ 85282
siddharths@asu.edu

Abstract

The field of sequential decision making (SDM) captures a range of mathematical frameworks geared towards the synthesis of goal-directed behaviors for autonomous systems. Abstract benchmark problems such as the *blocks-world domain* have facilitated immense progress in solution algorithms for SDM. there is some evidence that a direct application of SDM algorithms in real-world situations can produce unsafe behaviors. This is particularly apparent in task and motion planning in robotics. We believe that the reliability of today's SDM algorithms is limited because SDM models, such as the blocks-world domain, are *unsound* abstractions (those that yield false inferences) of real world situations.

This position paper presents the case for a focused research effort towards the study of sound abstractions of models for SDM and algorithms for efficiently computing safe goal-directed behavior using such abstractions.

Introduction

The increasing maturity of AI techniques presents us with a unique opportunity to develop physical and electronic AI agents that could autonomously assist humans. Such agents would need to be able to accept high-level commands, and reason about what to do over extended periods of time spanning multiple decision epochs. The field of sequential decision making (SDM) captures such problems. In order to solve them, an AI agent needs to the assess different courses of action available to it: which course of actions would accomplish the assigned task? would it be safe to execute? which course of actions would be beneficial? Evaluating a possible courses of action in this way requires some knowledge about the environment and the possible impacts of the agent's actions in it—in other words, *a model*.

In the absence of a model, such evaluations would need to be done through trial and error. It is difficult to conceive of situations where deploying robots would have a high value and where such trials and their associated errors would be acceptable. In situations that involve proximal human-robot collaboration, or situations that are too dangerous for humans, errors are usually associated with forbidding penalties. Just as a bomb-disposal robot that learns on the fly would be an ephemeral investment, a household assistant that attempts to learn through trial an error, which medication is required when a person goes into insulin shock, would be of dubious ethical, social and economic value. It is well known in the AI community that PAC-learning guarantees alone are not sufficient for ensuring safe behavior in such situations; recent analyses have highlighted their limitations in the face of the anticipated roles of AI systems (Russell, Dewey, and Tegmark 2015; Brynjolfsson and Mitchell 2017).

The focus of this position paper is on the *mechanisms for creating domain models that are efficient but sound abstractions of real-world problems, and the algorithmic advances required for using such models for safe behavior synthesis*. Models can be in the form of closed-form mathematical specifications, (such as Markov Decision Process with transition probability specifications) or in the form of black-box simulators or generative models that can sample possible action outcomes (as typically used in reinforcement learning). Models of either form can be derived from existing knowledge, or learned through past experience in the field. Indeed, some of the most popular demonstrations of AI systems rely upon *perfect models* (Silver et al. 2017; Mnih et al. 2015) in the form of game simulators for efficiently obtaining millions of labeled behavioral experiences.

Regardless of the form or the nature of acquisition of models, higher fidelity models feature larger branching factors and larger time horizons and therefore result in SDM problems of higher computational complexity (regardless of the solution approach taken, be it dynamic programming, search, learning from trials and past experience, or a combination thereof). Hierarchical abstractions are used to alleviate this problem by creating input models that are abstractions of the true problem (Sacerdoti 1974; Knoblock 1990; Parr and Russell 1998; Dietterich 2000; Marthi, Russell, and Wolfe 2007).

Hierarchical abstractions include state abstractions (models that maintain fewer environment properties than the real situation) as well as temporal abstractions (models featuring high-level actions that span multiple primitive operations of the underlying actuators).

In recent work (Srivastava, Russell, and Pinto 2016) we showed that simple forms of abstractions can result in models that are not consistent with the underlying problem scenario as well as models that are not Markovian, or not solvable! As a result many real-world problems have never truly been addressed by SDM solution techniques that treat their input models as perfect abstractions.

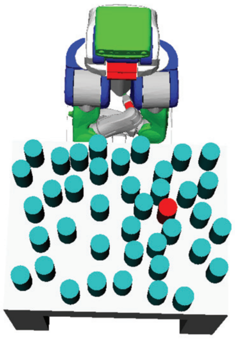


Figure 1: A realistic blocks-world problem. Pickups can be made only from the sides. Although there is no stacking and the preconditions of the pickup action are satisfied, there is no feasible motion plan for picking up most of the objects on the table.

For instance, consider the blocks-world domain, which is among the most easily recognizable, perhaps even infamous, benchmarks for sequential decision making. Given initial and desired configurations of blocks on a surface, the problem is to compute a behavior that would transform the initial configuration to the desired configuration. The set of available actions typically consists of maneuvers such as *pickup* and *place*. Stochastic action effects and noisy sensors for this domain can be easily expressed in most modeling languages for SDM (Boutilier, Reiter, and Price 2001; Younes and Littman 2004; Sanner 2010; Srivastava, Cheng, and Russell 2014). Although SDM *models* for the blocks-world domain are considered to be too "well studied" to be interesting for research, they are poor abstractions of the underlying SDM *problems* of rearranging objects while avoiding collisions (see Fig. 1 for a simplified yet realistic problem). Consequently the underlying problems remain unsolved and feature significant research challenges.

Indeed, while the true space of blocks-world problems captures all pick-and-place problems, ongoing research in robotics shows that SDM solvers that perform well on the standard blocks-world model produce poor *unsafe* solutions even in simplified real-world situations that feature robots with perfect sensing and actuation, and block arrangements without stacking (Cambon, Alami, and Gravot 2009; Kaelbling and Lozano-Pérez 2011; Plaku and Hager 2010; Kaelbling and Lozano-Pérez 2013; Srivastava et al. 2014). These solutions could violate arbitrarily many of the constraints that were abstracted in an unsound fashion, and result in unsafe behaviors that include unintended collisions. This situation is representative of several problems where goal-directed autonomy is desired; we believe that this potential for unsafe behavior effectively prohibits the safe deployment of general purpose AI agents.

Problem with the Current Situation

Conventional modeling paradigm As noted above, it is well appreciated that abstraction is a useful mathematical tool for solving real-world SDM problems. The conventional wisdom along these lines is to use an abstract domain model with an SDM solver to compute the "high-level strategy" for solving a problem (e.g., one that determines the order of unstacking and stacking block-tower configurations), and then use a low-level planner (e.g., a motion planner) or controller to implement each of the actions in the strategy.

Underlying assumptions and their limitations This wisdom is based on the assumption that the effect of applying an action in the real world will be consistent with the modeled effect in the abstract model. This in turn is based on the assumption that the result of applying a desired abstraction function on the real situation will be a Markovian model.

On the other hand, constructing a Markovian model requires the inclusion of several properties of the environment as state variables or predicates; abstraction requires *removing, or coarsening properties* in the model. It should therefore be natural to expect the abstraction of an accurate domain model to possibly result in a *non-Markovian domain model*. Recent work shows that this intuition is in fact true (Srivastava, Russell, and Pinto 2016): simple abstractions can result in models that are not Markovian; furthermore, it is often not possible to express the resulting models accurately in existing modeling languages for SDM.

This raises a few questions: all the SDM models we use are Markovian (and naturally, are expressed in the modeling languages that we have been using). Few, if any, of these are accurate, non-abstracted depictions of the real world situation that they represent. Have we been lucky enough to always get Markovian abstractions? Do the domain designers intuitively construct perfect abstract transition systems that retain just the right properties to make the resulting abstracted model tractable as well as Markovian?

To answer these questions, we turn once again to the blocks world and its abstraction expressed as the blocks-world domain. Among other details, this domain states that if a block has nothing on top of it, the robot's gripper should be able to pick it up. In a real situation (e.g. Fig. 1), this is *not true* because there may be no collision-free path for the gripper to pick up the block. The vocabulary used in the blocks-world domain is not sufficient to accurately express this property (Cambon, Alami, and Gravot 2009; Hertle et al. 2012; Kaelbling and Lozano-Pérez 2011). *As a result the standard blocks-world domain is not a sound model of the real blocks world because it implies action consequences that are not true* [1]. Policies computed using such models are unsafe, and can be dangerous. Although our example refers to situations where geometric constraints were abstracted out, such errors can arise with all forms of abstraction. One would not appreciate a robot using such principles in most applications that could benefit from a safe and productive robot, including mining, firefighting, bomb disposals, household help, etc.

[1] It is sound for environments where the gripper is either infinitesimally thin (so that it can slip between adjacent towers), or is an electromagnet suspended from the ceiling. Either way, the ceiling should be arbitrarily high and the table should be broad enough to lay any number of blocks on it. Such situations are unusual if not impossible.

In fact, the sound abstraction of the blocks world using the vocabulary of the blocks-world domain is a non-Markovian transition system: the effect of reaching for a block in this transition system depends on the occurrence of preceding *place* actions. If the target block was initially reachable, and no other *place* actions placed a block on the same table, the block will remain accessible. Otherwise, it may not be. *Therefore, the standard blocks-world model is not only inconsistent with the underlying problem, its vocabulary is insufficient to make the abstract transition system Markovian!* Forcing such a non-Markovian abstract transition system into domain languages that can only express Markovian models results in a model that is inconsistent with the modeled problem. Our research indicates that the situation can be resolved if the modeling languages are extended to annotate parts of the model as imprecise due to abstraction, and algorithms utilize this information to extract more information from higher fidelity models when needed.

Non-solutions The preceding discussion may *seem to indicate* that a stochastic formulation (such as an MDP) would help resolve these issues. However, this is not true. First, it would require enumerating and solving for the complete set of *possible* outcomes for an action in an abstract state space. This is infeasible. E.g., in the blocks-world model's vocabulary, every time a robot (not a ceiling mounted gripper) tries to move its hand, all possible subsets of movable objects in the room would need to be considered as potentially being knocked over. Second, such models would not be *complete*: they would disallow solutions that are feasible under a more accurate representation.

The problems highlighted above are orthogonal to efforts aimed at increasing the level of detail expressible in our input modeling languages (e.g., (Hertle et al. 2012; Fox and Long 2002)). Even if we could model SDM problems at the level of detail of sub-atomic particle interactions, this is unlikely to yield more efficient solution techniques. It is equally unlikely that modeling an entire household at the level circuit diagrams of every appliance would "help" a household robot efficiently compute useful behavior. Natural computational consequences of increasing branching factors and time horizons make it clear that a uniformly detailed model at the highest possible fidelity will not yield the most efficient SDM system, regardless of the solution approach. Thus, SDM solvers will continue to rely upon hierarchical abstractions for efficiency in modeling and in solution computation.

Paths Ahead

We believe that the limitations in correctly expressing abstract SDM models of real-world situations (and consequently, of efficiently solving such problems) have limited the applicability of SDM techniques in the real world. As a community we have made numerous advances under the assumption that inputs will be perfect abstractions that yield exactly the true consequences. Our position is that these advances are necessary, but not sufficient towards deployable autonomous systems. We also need to expand the scope of SDM technology towards principled approaches for designing and computing abstract SDM models that may be imprecise, but not incorrect. New representations for such abstract SDM models (generative models or *simulators*, as well as analytical) would require and facilitate corresponding algorithms that produce truly executable solutions.

Some prior research efforts are highly relevant to this problem. Work on algorithms for planning with models that may be incomplete addresses situations when unknown perturbations may have been applied to accurate domain models that are expressible in the modeling language (Nguyen and Kambhampati 2014). Angelic semantics for high-level actions increase the scope of representation languages to specify upper and lower bounds on reachable states in situations with temporal abstraction rather than state abstraction. The resulting algorithms are able to effectively utilize such bounds in pruning irrelevant high-level actions (Marthi, Russell, and Wolfe 2007). Related research in motion planning highlights the value of state abstractions of control-theoretic models, which are constructed using subsets of the full set of variables required to describe a system (Styler and Simmons 2017). We have been developing representations for efficiently expressing imprecise but sound abstract models resulting from state and temporal abstraction for arbitrary SDM problems. Our solution algorithms utilize sound and imprecise abstract models, but dynamically improve them by deriving abstracted, context-sensitive information from more accurate models. This information is abstracted and incorporated in the abstract models (Srivastava et al. 2014; Srivastava, Russell, and Pinto 2016), allowing SDM algorithms to compute agent behaviors with strong guarantees of safety and correctness. Some of our main results can be summarized as follows:

1. Under certain conditions, abstraction can indeed result in Markovian models. These conditions appear to be rare.
2. In many cases, abstraction results in domain models that includes forms of model-imprecision that could have been resolved during computation had they been expressed. However, current modeling languages do not support constructs that distinguish model imprecision arising due to abstraction from non-determinism or stochasticity that is a feature of the environment.
3. If model imprecision caused due to abstraction is recorded in the abstract model (e.g., by noting that the effect of a *place* action is imprecise, along with the abstraction function that caused the imprecision), the situation can be resolved. It is possible to dynamically tune the abstraction to include more information from accurate models using different solvers for models at different levels of abstraction. Used in this fashion, SDM solvers can effectively produce executable behavior. Dynamically tuning an imprecise (but not incorrect) model during search allowed us to produce a competitive task and motion planner that uses existing SDM solvers.

These initial results indicate that new methods for computing and utilizing abstract models that are sound even when they are imprecise allow us to leverage SDM technology towards solving entire new classes of problems that are abstractions of real-world situations.

Acknowledgments

I thank Rao Kambhampati for helpful discussions and comments on this paper.

References

Boutilier, C.; Reiter, R.; and Price, B. 2001. Symbolic dynamic programming for first-order mdps. In *Proc. IJCAI*, volume 1, 690–700.

Brynjolfsson, E., and Mitchell, T. 2017. What can machine learning do? workforce implications. *Science* 358(6370):1530–1534.

Cambon, S.; Alami, R.; and Gravot, F. 2009. A hybrid approach to intricate motion, manipulation and task planning. *IJRR* 28:104–126.

Dietterich, T. G. 2000. Hierarchical reinforcement learning with the maxq value function decomposition. *J. Artif. Intell. Res.(JAIR)* 13:227–303.

Fox, M., and Long, D. 2002. PDDL+: Modeling continuous time dependent effects. In *Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space*.

Hertle, A.; Dornhege, C.; Keller, T.; and Nebel, B. 2012. Planning with semantic attachments: An object-oriented view. In *Proc. ECAI*.

Kaelbling, L. P., and Lozano-Pérez, T. 2011. Hierarchical task and motion planning in the now. In *Proc. ICRA*.

Kaelbling, L. P., and Lozano-Pérez, T. 2013. Integrated task and motion planning in belief space. *The International Journal of Robotics Research* 32(9-10):1194–1227.

Knoblock, C. A. 1990. Learning abstraction hierarchies for problem solving. In *Proc. AAAI*.

Marthi, B.; Russell, S. J.; and Wolfe, J. 2007. Angelic semantics for high-level actions. In *Proc. ICAPS*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Nguyen, T. A., and Kambhampati, S. 2014. A heuristic approach to planning with incomplete STRIPS action models. In *ICAPS*.

Parr, R., and Russell, S. J. 1998. Reinforcement learning with hierarchies of machines. In *Proc. NIPS*.

Plaku, E., and Hager, G. D. 2010. Sampling-based motion and symbolic action planning with geometric and differential constraints. In *Proc. ICRA*.

Russell, S.; Dewey, D.; and Tegmark, M. 2015. Research priorities for robust and beneficial artificial intelligence. *AI Magazine* 36(4):105–114.

Sacerdoti, E. D. 1974. Planning in a hierarchy of abstraction spaces. *Artificial intelligence* 5(2):115–135.

Sanner, S. 2010. Relational dynamic influence diagram language (rddl): Language description. http://users.cecs.anu.edu.au/~ssanner/IPPC_2011/RDDL.pdf.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; Chen, Y.; Lillicrap, T.; Hui, F.; Sifre, L.; van den Driessche, G.; Graepel, T.; and Hassabis, D. 2017. Mastering the game of go without human knowledge. *Nature* 550(7676):354–359.

Srivastava, S.; Fang, E.; Riano, L.; Chitnis, R.; Russell, S.; and Abbeel, P. 2014. A modular approach to task and motion planning with an extensible planner-independent interface layer. In *Proc. ICRA*.

Srivastava, S.; Cheng, X.; and Russell, S. 2014. First-order open-universe POMDPs: Formulation and algorithms. In *Proc. UAI*.

Srivastava, S.; Russell, S.; and Pinto, A. 2016. Metaphysics of planning domain descriptions. In *Proc. AAAI*.

Styler, B. K., and Simmons, R. 2017. Plan-time multi-model switching for motion planning. In *Proc. ICAPS*.

Younes, H. L., and Littman, M. L. 2004. PPDDL 1.0: An extension to pddl for expressing planning domains with probabilistic effects. *Technical Report CMU-CS-04-162*.

Exploiting Micro-Clusters to Close The Loop in Data-Mining Robots for Human Monitoring

Einoshin Suzuki
Dept. Informatics, ISEE, Kyushu University
744 Motooka, Nishi, Fukuoka, 819-0395 Japan

Abstract

This paper describes our approach to integrating representation, reasoning, learning, and execution in our data-mining robots by exploiting micro-clusters to close the loop of the KDD process model. Based on our several kinds of autonomous mobile robots that monitor humans with Kinect and discover patterns, we are working on designing data-mining robots, each of which makes trials and errors in its data observation, data processing, pattern extraction, and mobile explorations. In other words, the robots continuously refine their goals at the micro-cluster level. We briefly discuss our four research directions, i.e., the balance between the exploitation and the exploration, the use of weak labels, the anytime algorithm, and the countermeasure to the concept drift, and describe potential, promising approaches for some of them.

Data-Mining Robots for Human Monitoring

We have constructed several kinds of autonomous mobile robots that monitor humans with Kinect and discover patterns. For instance, one to three robots, either a TurtleBot 2 or a hand-crafted robot each with Kobuki, jointly monitor a walking human, typically with elderly-experience equipment, to discover fall risks by clustering his/her skeletons (Deguchi et al. 2017; Takayama et al. 2014). Another example is a TurtleBot 2 with Kobuki that clusters facial expressions to discover smiling, yawning, and reading clusters of a desk worker (Kondo, Deguchi, and Suzuki 2014). This robot was later used to detect his/her hidden fatigue by clustering classifiers of neutral faces and smiling faces, which were observed every 30 minutes with their weak class labels input through a wireless mouse (Deguchi and Suzuki 2015). Figure 1 shows snapshots of these robots in the respective series of experiments.

All these robots represent the monitored person with micro clusters, which are learnt based on procedures similar to BIRCH, a hierarchical clustering algorithm (Zhang, Ramakrishnan, and Livny 1997; Han, Kamber, and Pei 2012). A micro cluster, which represents a group of similar examples each described with a set of numerical features, in its original form is a triplet $(n, \mathbf{v}, \mathrm{s})$, where n, $\mathbf{v}$, and s respectively represent the number of examples in the micro cluster, the add-sum of the examples in the micro cluster, and the add-sum of the squared L2-norm of the examples in the micro cluster (Zhang, Ramakrishnan, and Livny 1997). This triplet is called a Clustering Feature (CF) vector and has virtues of enabling an exact, incremental update and a reproduction of various cluster-wise distances without using the original examples. We initially adopted this approach to cluster colors of subimages observed by an autonomous mobile robot (Suzuki, Matsumoto, and Kouno 2012), and then extended the idea to cluster skeletons (Deguchi et al. 2017; Takayama et al. 2014), facial expressions (Kondo, Deguchi, and Suzuki 2014), and linear classifiers (Deguchi and Suzuki 2015). In these applications, an example is represented by a point in an Euclidean space spanned by the vectors of features, e.g., instability features described with skeleton joints inferred by Kinect (Deguchi et al. 2017; Takayama et al. 2014), action units inferred by Kinect to code emotional facial expressions (Kondo, Deguchi, and Suzuki 2014), coefficients of a logistic repression classifier to discriminate between neutral faces and smiling faces (Deguchi and Suzuki 2015).

Currently, we are working on extending our robots to data-mining robots, each of which makes trials and errors in its data observation, data processing, pattern extraction, and mobile explorations. The idea comes from the Knowledge Discovery in Databases (KDD) process model (Fayyad, Piatetsky-Shapiro, and Smyth 1996) shown in Figure 2. The Knowledge Discovery in Databases (KDD) process model states that a data mining process can be modeled as a series of several kinds of pre-/post-processing and pattern extraction. Our application domain is on a TurtleBot with Kobuki equipped with Kinect ver. 2 that continuously navigates inside a 90-m^2 room, observes desk workers, report discovered patterns to them, and receives their comments as rewards through its mouse. We believe that our data-mining robots are still goal-oriented, though their goals are unclear at the pattern level during their operations due to the nature of the KDD process model.

Exploiting Micro-Clusters to Close The Loop

Our previous robots either neglect the discovered patterns and micro-clusters or use them through static proce-

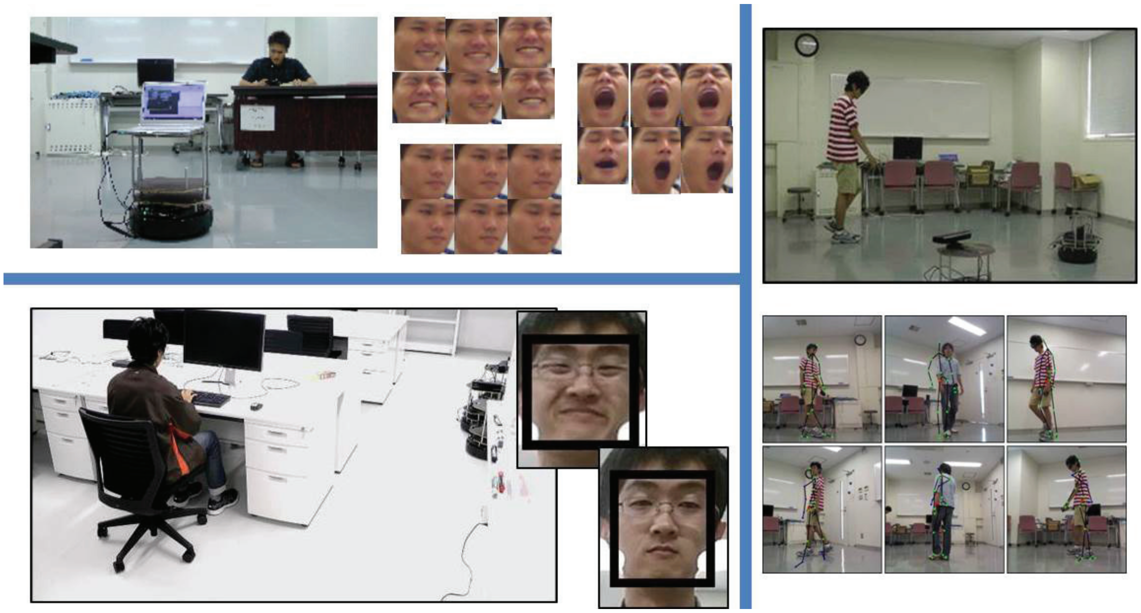


Figure 1: Snapshots of our autonomous mobile robots that monitor humans with Kinect and discover patterns. (Top left) TurtleBot 2 with Kobuki clusters facial expressions to discover smiling, yawning, and reading clusters of a desk worker (Kondo, Deguchi, and Suzuki 2014). (Right) Two TurtleBots 2 with Kobuki jointly monitor a walking human with elderly-experience equipment to discover fall risks by clustering his/her skeletons (Deguchi et al. 2017; Takayama et al. 2014). (Bottom left) TurtleBot 2 with Kobuki detects hidden fatigue of a desk worker by clustering classifiers of neutral faces and smiling faces, which were observed every 30 minutes with their weak class labels input through a wireless mouse (Deguchi and Suzuki 2015).

dures (Deguchi et al. 2017; Takayama et al. 2014; Kondo, Deguchi, and Suzuki 2014; Deguchi and Suzuki 2015). On the other hand, our intended data-mining robots closes “The Loop”, i.e., realizes the trials and errors of the KDD process model especially by exploiting their results of the pattern discovery in their data observation and mobile explorations. In other words, the robots continuously refine their goals at the micro-cluster level. We have adopted four research directions: the balance between the exploitation and the exploration, the use of weak labels, the anytime algorithm, and the countermeasure to the concept drift.

Realizing the balance between the exploitation and the exploration requires care in our application due to the difficulty in estimating the interestingness of a discovered pattern in data mining. Though we have already built naive methods, e.g., moving to observe from a different angle when the set of micro clusters reaches a pre-defined degree of stability, the reward given by humans is not necessarily related to such diversity and how to estimate the correct, new angle for observation is unclear. Note that we are mostly faced with signal data, as the symbol grounding problem is far from being resolved. Modeling the diversity related to the interestingness would be the next step, though the exploration for new data would remain hard-wired.

We define a weak label as a piece of information related with supervisory signal, or the desired output value. It could be a class label of a bag of examples in the multiple instance learning, a class label in relevant learning tasks in multitask or transfer learning, a (probabilistic) constraint on the target class labels in classification. See for instance (Mann and McCallum 2010). In our problem, the reward by a desk workers is rarely given, even if our robot reports an interesting pattern. We have recently developed a one-class selective transfer machine for personalized anomalous facial expression detection (Fujita, Matsukawa, and Suzuki 2018), which would be useful in both designing how to exploit weak labels and using the detected anomalous facial expressions as weak labels.

Naturally, our robot has to adopt an anytime algorithm, e.g., (Ueno et al. 2006), which can return the so-far best output anytime by using the available resources, especially the computation time. In BIRCH (Zhang, Ramakrishnan, and Livny 1997; Han, Kamber, and Pei 2012) and our discovery robots (Deguchi et al. 2017; Takayama et al. 2014; Kondo, Deguchi, and Suzuki 2014; Deguchi and Suzuki 2015), the micro-clusters are managed by a Clustering Feature (CF) tree, which may be viewed as a result of hierarchical clustering (Han, Kamber, and Pei 2012). Handling and reporting the micro-clusters in an intermediate level of the CF tree is a naive but natural solution. The closing the loop problem dictates that this research direction is deeply related with the first one: the balance between the exploitation and the exploration. Combined with the other two problems, designing an adequate anytime algorithm for our robots raises numerous challenges, even if partial solutions exist in the literature, e.g., (Ivanov, Blumberg, and Pentland: 2001).

Last but not least, our robot has to take a countermeasure to the concept drift, which is inherent in data stream

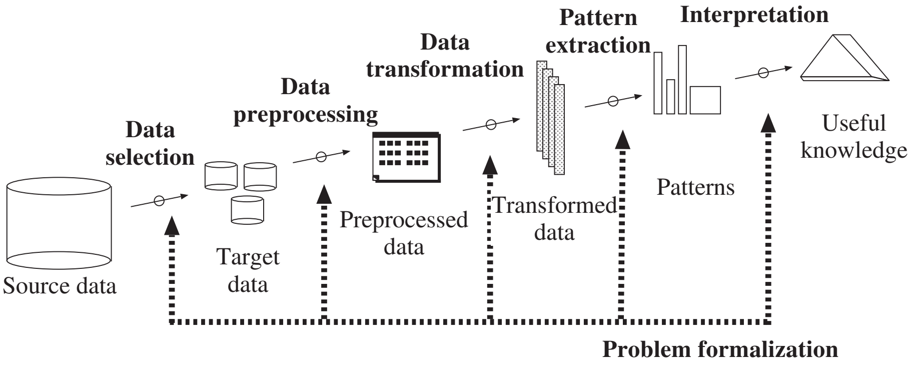


Figure 2: KDD process model (adopted and modified from (Fayyad, Piatetsky-Shapiro, and Smyth 1996)).

mining (Krempl et al. 2014). The statuses of desk workers change gradually or abruptly, though our robot platform including its batteries and sensors is reliable and can be regarded as static. Comparing CF trees (Boubou, Hafez, and Suzuki 2015) is in fact a nontrivial procedure and thus we are rather seeking for another approach of managing a set of micro-clusters.

Acknowledgments

I deeply appreciate all co-authors and collaborators. A part of this research was supported by Grant-in-Aid for Scientific Research JP24650070, JP25280085, and JP15K12100 from the Japan Society for the Promotion of Science (JSPS); a Bilateral Joint Research Project between Japan and France funded by JSPS and CNRS (CNRS/JSPS PRC 0672); and Strategic International Cooperative Program funded by Japan Science and Technology Agency (JST) and ANR.

References

Boubou, S.; Hafez, A. H. A.; and Suzuki, E. 2015. Visual Impression Localization of Autonomous Robots. In *Proc. 2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 328–334.

Deguchi, Y., and Suzuki, E. 2015. Hidden Fatigue Detection for a Desk Worker Using Clustering of Successive Tasks. In *Ambient Intelligence*, volume 9425 of *LNCS*, 263–238. Springer-Verlag.

Deguchi, Y.; Takayama, D.; Takano, S.; Scuturici, V.-M.; Petit, J.-M.; and Suzuki, E. 2017. Skeleton Clustering by Multi-Robot Monitoring for Fall Risk Discovery. *Journal of Intelligent Information Systems* 48(1):75–115.

Fayyad, U. M.; Piatetsky-Shapiro, G.; and Smyth, P. 1996. From Data Mining to Knowledge Discovery: An Overview. In *Advances in Knowledge Discovery and Data Mining*. Menlo Park, Calif.: AAAI/MIT Press. 1–34.

Fujita, H.; Matsukawa, T.; and Suzuki, E. 2018. One-Class Selective Transfer Machine for Personalized Anomalous Facial Expression Detection. In *Proc. Thirteenth International Conference on Computer Vision Theory and Applications (VISAPP)*. (accepted for publication).

Han, J.; Kamber, M.; and Pei, J. 2012. *Data Mining, Concepts and Techniques*. Morgan Kaufmann.

Ivanov, Y. A.; Blumberg, B.; and Pentland:, A. 2001. Expectation Maximization for Weakly Labeled Data. In *Proc. ICML 2001*, 218–225.

Kondo, R.; Deguchi, Y.; and Suzuki, E. 2014. Developing a Face Monitoring Robot for a Deskworker. In *Ambient Intelligence*, volume 8850 of *LNCS*, 226–241. Springer-Verlag.

Krempl, G.; Žliobaite, I.; Brzeziński, D.; Hüllermeier, E.; Last, M.; Lemaire, V.; Noack, T.; Shaker, A.; Sievi, S.; Spiliopoulou, M.; and Stefanowski, J. 2014. Open Challenges for Data Stream Mining Research. *SIGKDD Explorations* 16(1):1–10.

Mann, G. S., and McCallum, A. 2010. Generalized Expectation Criteria for Semi-Supervised Learning with Weakly Labeled Data. *Journal of Machine Learning Research* 11:955–984.

Suzuki, E.; Matsumoto, E.; and Kouno, A. 2012. Data Squashing for HSV Subimages by an Autonomous Mobile Robot. In *Discovery Science (DS)*, volume 7569 of *LNAI*, 95–109. Springer-Verlag.

Takayama, D.; Deguchi, Y.; Takano, S.; Scuturici, V.-M.; Petit, J.-M.; and Suzuki, E. 2014. Multi-view Onboard Clustering of Skeleton Data for Fall Risk Discovery. In *Ambient Intelligence*, volume 8850 of *LNCS*, 258–273. Springer-Verlag.

Ueno, K.; Xi, X.; Keogh, E. J.; and Lee, D.-J. 2006. Anytime Classification Using the Nearest Neighbor Algorithm with Applications to Stream Mining. In *Proc. ICDM 2006*, 623–632.

Zhang, T.; Ramakrishnan, R.; and Livny, M. 1997. BIRCH: A New Data Clustering Algorithm and its Applications. *Data Mining and Knowledge Discovery* 1(2):141–182.

Learning Abstractions by Transferring Abstract Policies to Grounded State Spaces

Lawson L. S. Wong
Department of Computer Science, Brown University
Providence, RI 02912, USA
lsw@brown.edu

Abstract

Learning from demonstration is an effective paradigm to teach specific tasks to robots. However, such demonstrations often have to be performed on the robot, which is both time-consuming and often still requires expert knowledge (e.g., kinesthetically controlling the joints). It is often easier to specify tasks at a high level of abstraction, and let the robot figure out the grounding to the robot/agent space. We consider how to learn such a mapping. In particular, we consider the task of learning to navigate on a mobile robot given only an abstraction of the path and potential landmarks. We cast this as a learning problem between abstract and robot (grounded) state spaces and illustrate how this works in several cases. Through these cases, we see that the "abstract navigation" task touches on many interesting issues related to abstraction, and suggest avenues for further investigation.

Introduction

To tackle the high-dimensional complexity of the world and long-horizon nature of complex tasks, agents need *abstraction*, the act of compressing both state and time in service of certain goals. Much of artificial intelligence has been devoted to manually endowing agents with abstractions, such as via symbols (state abstraction) (Dietterich 2000; Konidaris, Kaelbling, and Lozano-Pérez 2018) and subtasks/options (temporal abstraction) (Sutton, Precup, and Singh 1999). However, agents that operate in a continual and lifelong setting will eventually encounter conditions unforeseen to the designer, and must come up with its own abstractions. Existing work in learning abstractions, most notably in reinforcement learning, typically require much experience within the domain, and arguably have not achieved widespread success. Indeed, one of the challenging aspects of abstraction is that in the time it takes to induce an abstraction and learn how to use it effectively, the specific ground / non-abstract task could already have been solved.

In contrast, humans use abstractions very effectively. For example, when provided a 2-D map of a new location (e.g., Figure 1), people can typically follow the map to reach a desired destination on the first try, without requiring the numerous episodes of trial and error that reinforcement learners require. This feat is even more remarkable when considering that the real-world looks nothing like the 2-D map: it is 3-D, is perceived from a first-person perspective (instead of bird's-eye for maps), and contains many more objects and other distractors compared to the map itself. Even so, when encountering these completely new percepts and 'states', people can follow where they are on the map and navigate as desired. Humans have mastered the abstraction of 2-D maps: from the current *ground* state in the real world, they are able to find the corresponding *abstract* state as a 2-D point on the map, determine the appropriate next *abstract* action within the abstract world, and then *ground* this action into physical motion. Furthermore, humans have mastered the entire *class* of such 2-D map abstractions; given a *new* instance of the abstraction (e.g, a map of a new place), humans can immediately perform the necessary grounding.

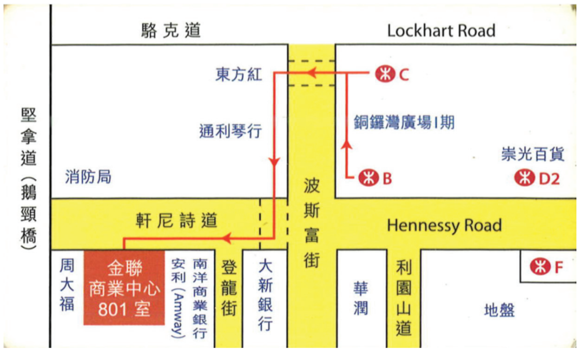


Figure 1: Humans can navigate in new places by using abstract 2-D maps, such as by following the walking directions depicted by the red arrows in the map above, which direct a person to exit a certain subway exit and cross two roads to reach an office (red square in the bottom left). They are able to take *abstract* policy-related knowledge encoded in the map, and *ground* the relevant actions in the real world. If robotic agents can learn to use existing abstractions, not only will they be easier to instruct by humans, they may even be able to produce abstract and interpretable knowledge.

We first formalize the notion of abstraction, then frame the problem of learning how to use existing abstractions as a fully supervised, learning-from-demonstration problem. For the "abstract navigation" task described above, we consider several classes of possible abstractions, some of which are

$$\begin{tikzcd}
S \arrow[r, "\pi"] \arrow[d, "f_s"'] & A^+ \\
\widetilde{S} \arrow[r, "\tilde{\pi}"] & \widetilde{A} \arrow[u, "f_a"']
\end{tikzcd}$$

Figure 2: Abstraction diagram.

easy to learn, whereas others remain unsolved. Finally, we discuss directions of ongoing and future investigation.

Related Work

The general problem setup has ties to transfer learning (Taylor and Stone 2009) and learning from demonstration (Argall et al. 2009). Cobo et al. (Cobo et al. 2014) also explored learning abstractions from demonstrations, using an approach based on feature selection and task decomposition.

The formulation of abstractions in this work is inspired by the pioneering work of Ravindran (Ravindran 2004) and subsequent work by Abel et al. (Abel, Hershkowitz, and Littman 2016). Both lines of work analyze theoretical properties of abstraction in reinforcement learning.

Recently, there has been work on navigation using abstract 2-D maps such as hand-sketched maps (Boniardi et al. 2015; 2016), floor plans (Gao et al. 2017), and mazes (Brunner et al. 2018). However, most of these approaches are specific to 2-D robot/agent navigation.

Model and Problem Formulation

The agent operates in the grounded state space S and action space A. The objective is to determine a plan or policy $\pi : S \rightarrow A$ that achieves some given task in the world. The premise of this work is that we are given an abstract solution for the task, such as a route to follow on an abstract 2-D map that reaches a desired goal location. Formally, we are given an abstraction in abstract state and action spaces $\widetilde{S}, \widetilde{A}$, as well as an abstract policy $\tilde{\pi} : \widetilde{S} \rightarrow \widetilde{A}$.

The abstract policy is a solution for the (grounded) task if there exist *abstraction functions* $f_s : S \rightarrow \widetilde{S}$ and $f_a : \widetilde{A} \rightarrow A^+$ that can produce a grounded policy according to the diagram in Figure 2. In particular, the requirement is:

$$\pi = f_a \circ \tilde{\pi} \circ f_s \qquad (1)$$

To find the next ground action(s), we first lift the ground state to the abstract state using f_s, apply the given abstract policy $\tilde{\pi}$, then ground the resulting abstract action using f_a to an executable primitive action (or action sequence, if there is temporal abstraction). If $\tilde{\pi}$ is an abstract solution for the task, then repeatedly applying this procedure should result in the agent reaching the goal in its grounded space.

Our goal is to *learn* the abstraction functions f_s and f_a, such that when presented with a new instance of the abstraction class (e.g., a 2-D map of a new location), the agent can follow the given abstract solution via Equation 1, i.e., transfer an abstract policy to the agent's grounded state space.

To learn the abstraction functions, we need training data. We consider the simplest setting, where paired trajectories in both ground and abstract spaces are provided. This is a fully-supervised, learning-from-demonstration setting, where the agent is shown grounded solutions to various task instances (e.g., by guiding it through the real world), together with annotated abstract solutions to the same problems (e.g., by drawing the route on the 2-D map).

Abstract Navigation

We consider several instances of a problem where the task is to follow a specified path, given in an abstract space. The grounded state space in all these cases is the state of the robotic agent, which includes highly relevant state dimensions such as odometry (noisy estimate of location relative to its starting position), moderately relevant features such as detected landmarks, and irrelevant features such as its arms' joint angles (if it has arms) or its battery level.

Isometric path

In the simplest case, the abstract path is given as a 2-D trajectory that accurately preserves relative lengths and angles, except possibly in a different global coordinate frame and scale. (This would be the case if the path was specified in most popular web mapping services such as Google Maps.) If the abstract path is also annotated at each point with the appropriate ground action, which could also be easily inferred from an isometric 2-D solution trajectory, then f_a can be assumed to be the identity function. The paired trajectories during training give corresponding pairs $(s, \tilde{s})$ of high-dimensional ground states and 2-D abstract states respectively. Learning f_s then becomes a multi-label linear regression problem (mapping s to $\tilde{s}$), since the ground and abstract states are related via an affine transformation (in the case of an isometric abstract path). In simulation, this method alone is highly effective at ignoring irrelevant features in the ground state s and handling zero-mean additive noise.

For abstract paths that are not perfect isometries, we need to learn non-linear regression functions. This is still strictly within the realm of supervised machine learning, for which many approaches exist to learn non-linear f_s functions.

The issue of orientation

The previous case provided a way to accurately find the abstract (x, y) location on the provided abstract path. However, the first problem one encounters when implementing the strategy on a point robot is orientation: if the robot is not facing in the same direction as the path intended, then following the abstract policy $\tilde{\pi}$ causes the robot to deviate from the path. The main issue is that the abstraction is insufficient to distinguish between the canonical path-following orientation from other states sharing the same abstract (x, y).

There are several potential ways to fix this. The simplest is to expand the abstract space to incorporate orientation θ as well; however, this requires a more complicated abstract policy to be specified. Alternatively, the burden may be placed on the agent, by formulating each step of the path-following as a subtask (instead of a primitive action), where the subgoal is to return to a canonical orientation. The canonical orientation can be learned during training, or may be required to be the initial heading of the robot.

More generally, incomplete abstractions are likely to be encountered, and it would be useful to detect them and make local corrections, such as by inserting subgoals. This points to one argument for learning both ground and abstract transition models, T and $\widetilde{T}$ respectively: an incomplete abstraction will not generally be able to enforce one-step consistency between $f_s \circ T$ and $\widetilde{T} \circ f_s$. Thus transition models enable error detection in abstraction.

Landmarks

In typical maps, even in the case that the map is an isometry, there are additional features such as street names, room numbers, and other iconic elements such as architecturally distinct buildings. For example, in Figure 1, various street names (black font) and store names (blue font) are given near their respective locations. As humans, our sense of odometry is likely worse than mobile robots, so we must rely on these highly distinguishable landmark cues for robustness. If the robot is provided with detectors that allow it to detect landmark features, then these detections can simply be incorporated as additional ground state dimensions, and we can proceed to learn f_s from demonstrations via nonlinear regression. In simulation, we considered landmarks in the form of 'color patches' encountered in local regions of the world; if the color is confined to a unique region in the abstract space, these landmarks are highly informative and can correct for otherwise inaccurate geometric mappings.

Topological path

In the previous case, landmarks provide information that is redundant with the geometric abstract map. Hence it is possible to remove the geometric aspects of the abstraction and simply retain the topological information provided by landmarks. A path in the space of landmarks can now be represented as a deterministic finite automaton; for example, in the case of street names as landmarks, nodes may correspond to streets, and edges with street intersections (with an appropriate output ground action to perform the correct turn, if any). Note that this abstraction only allows specifying a single action to be repeatedly performed between two landmarks; for example, when on a certain street, the agent can only move in one direction on the street, until an intersection is encountered. In this case, uniqueness of landmarks is essential, since they are the only source of information, unless transition models are also provided to enable tracking.

Richer abstractions

The initial motivation for the abstract mapping task was to follow an abstract 2-D map, such as the one in Figure 1. Ultimately, these maps are typically perceived via vision, and it would be much easier for a robot to use existing maps if it can process them in image form, rather than requiring a manual encoding of the abstract policy $\tilde{\pi}$. Compared to previous cases, using the 2-D map in image form is interesting because it is both featurally richer compared to previous abstractions, while at the same time still much lower-dimensional with respect to the robot. One possibility for using this image-based abstraction is to extract features from it, such as using convolutional neural networks, and to then learn to map ground states to abstract visual features.

The automaton-based abstraction in the previous case is also closely related to using natural language instructions for navigation. For example, "go straight on street A for two blocks until the intersection with street B, then turn left" can be represented as an automaton. We can therefore consider using natural language itself as an abstraction, either by mapping the sequence of instructions to an automaton, or by directly mapping ground states to abstract linguistic features, as in the case for images.

Discussion

We considered the problem of learning to use existing abstractions in novel environments, in the context of the problem of navigation using abstract 2-D maps. The problem was formulated as a fully-supervised, learning from demonstration problem, and several cases of potential abstraction classes were considered. In the process of analyzing these cases, various aspects and issues of abstraction were encountered, and many problems and solutions still lie ahead.

There remains the issue of learning the action abstraction function f_a. This is the problem of temporal abstraction, which has arguably received greater attention in the field thus far. One way to consider an abstract action $\tilde{a}$ is to view it as a subgoal, which instantiates a local planning problem. This was a potential strategy used to overcome the lack of orientation information in the abstract 2-D map.

So far, the problem has only been considered in the fully-supervised setting. Although this provides the strongest signal for learning, it also requires significant effort from the user. One possibility is provide weak supervision through reinforcement learning, in the extreme case only providing reward if the correct path is followed. An intermediate regime would be to still provide demonstrations, but no longer with ground-abstract state correspondences.

Two cases in the previous section touched upon the utility of learning transition models. The benefit so far appears to be increased robustness in determining the correct abstract state. Transition models are also needed for planning; if the solution path is not provided, and only an abstract map is given (which is the case when using a standard map), then planning in the abstract space will be necessary. This is a useful extension to the problem considered so far: find an abstract policy and follow it via the same grounding mechanism, with the assumption that the abstraction is a "faithful" representation of the world with respect to the task.

The proposed approach may also provide useful theoretical analysis of abstractions. Since the problem of learning abstractions has been transformed into one of supervised learning, we may be able to adapt theoretical tools from computational learning theory in this more familiar setting, and characterize the utility of various abstractions. In particular, to use the given abstraction effectively, we had to learn the abstraction function f_s (and eventually f_a); the complexity of this learning problem tells us how practical the abstraction is. If it is difficult to learn f_s, then it may not be worth the extra learning effort for the potential reduction in representational complexity. An abstraction may only be useful if it is an accurate representation of the world with respect to the task, provides some degree of information compression, *and the abstraction functions are easy to learn*.

References

Abel, D.; Hershkowitz, D.; and Littman, M. 2016. Near optimal behavior via approximate state abstraction. In *International Conference on Machine Learning*.

Argall, B.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57(5):469–483.

Boniardi, F.; Behzadian, B.; Burgard, W.; and Tipaldi, G. 2015. Robot navigation in hand-drawn sketched maps. In *European Conference on Mobile Robots*.

Boniardi, F.; Valada, A.; Burgard, W.; and Tipaldi, G. 2016. Autonomous indoor robot navigation using a sketch interface for drawing maps and routes. In *IEEE International Conference on Robotics and Automation*.

Brunner, G.; Richter, O.; Wang, Y.; and Wattenhofer, R. 2018. Teaching a Machine to Read Maps with Deep Reinforcement Learning. In *AAAI Conference on Artificial Intelligence*.

Cobo, L.; Subramanian, K.; Isbell, C.; Lanterman, A.; and Thomaz, A. 2014. Abstraction from demonstration for efficient reinforcement learning in high-dimensional domains. *Artificial Intelligence* 216:103–128.

Dietterich, T. 2000. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research* 13:227–303.

Gao, W.; Hsu, D.; Lee, W.; Shen, S.; and Subramanian, K. 2017. Intention-net: Integrating planning and deep learning for goal-directed autonomous navigation. In *Conference on Robot Learning*.

Konidaris, G.; Kaelbling, L.; and Lozano-Pérez, T. 2018. From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research* 61:215–289.

Ravindran, B. 2004. *An Algebraic Approach to Abstraction in Reinforcement Learning*. Ph.D. Dissertation, University of Massachusetts Amherst.

Sutton, R.; Precup, D.; and Singh, S. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112(1):181–211.

Taylor, M., and Stone, P. 2009. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research* 10(1):1633–1685.

Information-Efficient Model Identification for Tensegrity Robot Locomotion

Shaojun Zhu, David Surovik, Kostas Bekris, Abdeslam Boularias
Department of Computer Science, Rutgers University, New Jersey, USA
{shaojun.zhu, david.surovik, kostas.bekris, abdeslam.boularias}@cs.rutgers.edu

Abstract

This paper aims to identify in a practical manner unknown physical parameters, such as mechanical models of actuated robot links, which are critical in dynamical robotic tasks. Key features include the use of an off-the-shelf physics engine and the data-efficient adaptation of a black-box Bayesian optimization framework. The task being considered is locomotion with a high-dimensional, compliant Tensegrity robot. A key insight in this case is the need to project the system identification challenge into an appropriate lower dimensional space. Comparisons with alternatives indicate that the proposed method can identify the parameters more accurately within the given time budget, which also results in more precise locomotion control.

Introduction

This paper presents an approach for model identification by exploiting the availability of off-the-shelf physics engines used for simulating dynamics of robots and objects they interact with. There are many examples of popular physics engines that are becoming increasingly efficient (Erez, Tassa, and Todorov, 2015; Bul; MuJ; DAR; Phy; Hav). These physics engines receive as input mechanical and mesh models of the robots in a particular scene, in addition to controls (force, torque, velocity, etc.) applied to them, and return a prediction of the robot's dynamical response.

The accuracy of the prediction depends on several factors. The first one is the limitation of the mathematical model used by the engine (e.g., the Coulomb approximation). The second factor is the accuracy of the numerical algorithm used for solving the equations of motion. Finally, the prediction depends heavily on the accuracy of the physical parameters of the robots, such as mass, friction and elasticity. In this work, we focus on the last factor and propose a method to improve the accuracy of the physical parameters used in the physics engine.

In the context of compliant locomotion systems, the Tensegrity robot of Figure 1 is a structurally compliant platform that can distribute forces into linear elements as pure compression or tension (Caluwaerts et al., 2014). This robot's tensile elements can be actuated, enabling it to effectively adapt to complex contact dynamics in unstructured terrains.

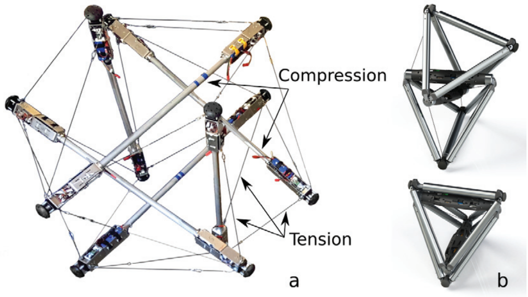


Figure 1: The Tensegrity robot (Caluwaerts et al., 2014).

A policy for a rolling locomotive gait of the platform has been learned from simulated data (Geng et al., 2016). Tensegrity robots are inherently high-dimensional, highly-dynamic systems, and providing a predictive model requires a physics-based simulator (NTRT). The accuracy of such a solution critically depends upon physical parameters of the robot, such as the density of its rigid elements and the elasticity of the tensile elements. While a manual process can be followed to tune a simulation to match the behavior of a real prototype (Mirletz et al., 2015), it is highly desirable to conduct this calibration using as few observed trajectories as possible. In this work, trajectories generated by a simulation manually tuned to a prototypical robotic platform are used to identify the parameters of a physics engine for tensegrity modeling. Given the high-dimensionality of the parameter space, this is a challenging problem. This work proposes the mapping of the system identification process to a lower dimensional space of parameters. Methods used for dimensionality reduction include Random Embedding (REMBO) (Wang et al., 2016) as well as Variational Auto Encoder (VAE) (Kingma and Welling, 2014). A data-efficient Bayesian optimization technique is used for searching in the lower dimensional space, instead of the original high dimensional parameter space. The proposed method is able to efficiently identify the parameters that produce a simulation that most closely matches the observed ground-truth trajectories of this exciting locomotive platform.

Foundations and Contributions

Two high-level approaches exist for learning robotic tasks with unknown dynamical models: model-free and model-based ones. Model-free methods search for a policy that best solves the task without explicitly learning the system dynamics (Sutton and Barto, 1998; Bertsekas and Tsitsiklis, 1996; Kober, Bagnell, and Peters, 2013; Levine and Abbeel, 2014). Model-free methods are accredited with the recent success stories of reinforcement learning in video games (Mnih et al., 2015). For robot learning, a relative entropy policy search has been used (Peters, Mülling, and Altün, 2010) to successfully train a robot to play table tennis. The PoWER algorithm (Kober and Peters, 2009) is another model-free policy search approach widely used in robotics.

Model-free methods, however, do not easily generalize to unseen regions of the state-action space. To learn an effective policy, features of state-actions in learning and testing should be sampled from distributions that share the same support. This is rather dangerous in robotics, as poor performance in testing could lead to irreversible damage.

Model-based approaches explicitly learn the dynamics of the system, and search for an optimal policy using standard simulation, planning, and actuation control loops for the learned parameters. There are many examples of model-based approaches for robotic manipulation (Dogar et al., 2012; Lynch and Mason, 1996; Merili, Veloso, and Akin, 2014; Scholz et al., 2014; Zhou et al., 2016), some of which have used physics-based simulation to predict the effects of pushing flat objects on a smooth surface (Dogar et al., 2012). A nonparametric approach was employed for learning the outcome of pushing large objects (furniture) (Merili, Veloso, and Akin, 2014). A Markov Decision Process (MDP) has been applied to modeling interactions between objects; however, only simulation results on pushing were reported (Scholz et al., 2014). For general-purpose model-based reinforcement learning, the PILCO algorithm has been proven efficient in utilizing a small amount of data to learn dynamical models and optimal policies (Deisenroth, Rasmussen, and Fox, 2011).

Bayesian Optimization is a popular framework for data-efficient black-box optimization (Shahriari et al., 2016). In robotics, some recent applications include learning controllers for bipedal locomotion (Antonova, Rai, and Atkeson, 2016), gait optimization (Calandra et al., 2016) and transfer policies from simulation to real world (Marco et al., 2017).

This work is based on a model-based approach, which instead of learning a dynamics model, it utilizes a physics engine, and concentrates on identifying only the mechanical properties of the objects instead of recreating the dynamics from scratch. Furthermore, it utilizes Bayesian optimization and identifies a process for dealing with high-dimensional system identification challenges efficiently.

Proposed Approach

This work proposes an online approach for robots to learn the physical parameters of their dynamics through minimal physical interaction. Because of the high dimensionality of the parameter space of the tensegrity robot, even with efficient optimization method like Bayesian optimization (BO), it is still challenging to identify all the parameters efficiently. The overall framework of the model identification process is first introduced, then the approaches of dimensionality reduction to decrease the search space of BO in order to achieve efficient optimization are covered in detail.

Model Identification

For the tensegrity robot, the physical properties of interest correspond to the density, length, radius, stiffness, damping factor, pre-tension, motor radius, motor friction, and motor inertia of the various rigid and tensile elements and actuators.

These physical properties are represented as a D-dimensional vector $\theta \in \Theta$, where Θ is the space of all possible values of the physical properties. Θ is discretized with a regular grid resolution. The proposed approach returns a distribution P on discretized Θ instead of a single point $\theta \in \Theta$ since model identification is generally an ill-posed problem. In other terms, there are multiple models that can explain an observed trajectory with equal accuracy. The objective is to preserve all possible explanations for the purposes of robust planning.

The online model identification algorithm (given in Algorithm 1) takes as input a prior distribution P_t, for time-step $t \geq 0$, on the discretized space of physical properties Θ. P_t is calculated based on the initial distribution P_0 and a sequence of observations $(x_0, \mu_0, x_1, \mu_1, \ldots, x_{t-1}, \mu_{t-1}, x_t)$. For the Tensegrity robot, x_t is a state vector concatenating the 3D centers of all rigid elements, i.e., the rods in the corresponding Figure 1, and μ_t is a vector of motor torques.

The process consists of simulating the effects of the controls μ_i on the robot in states x_i under various values of parameters θ and observing the resulting states $\hat{x}_{i+1}$, for $i = 0, \ldots, t$. The goal is to identify the model parameters that make the outcomes $\hat{x}_{i+1}$ of the simulation as close as possible to the real observed outcome x_{i+1}. In other terms, the following black-box optimization problem is solved:

$$\theta^* = \arg\min_{\theta \in \Theta} E(\theta) \overset{def}{=} \sum_{i=0}^{t} \|x_{i+1} - f(x_i, \mu_i, \theta)\|_2, \qquad (1)$$

wherein x_i and x_{i+1} are the observed states of the robot at times i and $i+1$, μ_i is the control that applied at time t, and $f(x_i, \mu_i, \theta) = \hat{x}_{i+1}$, the predicted state at time $t+1$ after simulating control μ_i at state x_i using physical parameters θ.

The proposed approach consists of learning the error function E from a sequence of simulations with different parameters $\theta_k \in \Theta$. To choose these parameters efficiently in a way that quickly leads to accurate parameter estimation, a belief about the actual error function is maintained. This belief is a probability measure over the space of all functions $E : \mathbb{R}^D \to \mathbb{R}$, and is represented by a Gaussian Process (GP) (Rasmussen and Williams, 2005) with mean vector m and covariance matrix K. The mean m and covariance K of the GP are learned from data points $\{(\theta_0, E(\theta_0)), \ldots, (\theta_k, E(\theta_k))\}$, where θ_k is a vector of physical properties of the object, and $E(\theta_k)$ is the accumulated distance between actual observed states and states that are obtained from simulation using θ_k.

The probability distribution P on the identity of the best physical model θ^*, returned by the algorithm, is computed

```
Input: State-action-state data $\{(x_i, \mu_i, x_{i+1})\}$ for
        $i = 0, \ldots, t$
        $\Theta$, a discretized space of possible values of
        physical properties;
Output: Probability distribution $P$ over $\Theta$ according to
        the provided data;
Sample $\theta_0 \sim \text{Uniform}(\Theta)$; $L \leftarrow \emptyset$; $k \leftarrow 0$;
repeat
    $l_k \leftarrow 0$;
    for $i = 0$ to $t$ do
        Simulate $\{(x_i, \mu_i)\}$ using a physics engine with
          physical parameters $\theta_k$ and get the predicted
          next state $\hat{x}_{i+1} = f(x_i, \mu_i, \theta_k)$ ;
        $l_k \leftarrow l_k + \|\hat{x}_{i+1} - x_{i+1}\|_2$;
    end
    $L \leftarrow L \cup \{(\theta_k, l_k)\}$;
    Calculate $GP(m, K)$ on error function $E$, where
      $E(\theta) = l$, using data $(\theta, l) \in L$;
    Sample $E_1, E_2, \ldots, E_n \sim GP(m, K)$ in $\Theta$;
    foreach $\theta \in \Theta$ do
        $P(\theta) \approx \frac{1}{n} \sum_{j=0}^{n} \mathbf{1}_{\theta = \arg\min_{\theta' \in \Theta} E_j(\theta')}$
    end
    $\theta_{k+1} = \arg\min_{\theta \in \Theta} P(\theta) \log\big(P(\theta)\big)$ ;
    $k \leftarrow k + 1$;
until *Timeout*;
```

Algorithm 1: Model Identification with Greedy Entropy Search

from the learned GP as

$$
\begin{aligned}
P(\theta) &\overset{def}{=} P\big(\theta = \arg\min_{\theta' \in \Theta} E(\theta')\big) \\
&= \int_{E:\mathbb{R}^D \to \mathbb{R}} p_{m,K}(E) \Pi_{\theta' \in \Theta - \{\theta\}} H\big(E(\theta') - E(\theta)\big) \mathrm{d}E
\end{aligned}
\tag{2}
$$

where H is the Heaviside step function, i.e., $H\big(E(\theta') - E(\theta)\big) = 1$ if $E(\theta') \geq E(\theta)$ and $H\big(E(\theta') - E(\theta)\big) = 0$ otherwise, and $p_{m,K}(E)$ is the probability of a function E according to the learned GP mean m and covariance K. Intuitively, $P(\theta)$ is the expected number of times that θ happens to be the minimizer of E when E is a function distributed according to GP density $p_{m,K}$.

Distribution P from Equation 2 does not have a closed-form expression. Therefore, a *Monte Carlo* sampling is employed for estimating P. Specifically, the process samples vectors containing values that E could take, according to the learned Gaussian process, in the discretized space Θ. $P(\theta)$ is estimated by counting the fraction of sampled vectors of the values of E where θ happens to have the lowest value, as indicated in Algorithm 1.

Finally, the computed distribution P is used to select the next vector θ_{k+1} to use as a physical model in the simulator. This process is repeated until the entropy of P drops below a certain threshold, or until the algorithm runs out of the allocated time budget. The entropy of P is given as $\sum_{\theta \in \Theta} -P_{min}(\theta) \log\big(P_{min}(\theta)\big)$. When the entropy of P is close to zero, the mass of distribution P is concentrated around a single vector θ, corresponding to the physical model that best explains the observations. Therefore, the next vector θ_{k+1} should be selected such that the entropy of P would decrease after adding the data point $\big(\theta_{k+1}, E(\theta_{k+1})\big)$ to train the GP and re-estimate P using the new mean m and covariance K in Equation 2.

The Entropy Search method (Hennig and Schuler, 2012) follows this reasoning and use Monte Carlo again to sample, for each potential choice of θ_{k+1}, a number of values that $E(\theta_{k+1})$ could take according to the GP in order to estimate the expected change in the entropy of P and choose the parameter vector θ_{k+1} that is expected to decrease the entropy of P the most. The existence of a secondary nested process of Monte Carlo sampling makes this method impractical for online model identification. Instead, this work proposes a simple heuristic for choosing the next θ_{k+1}. In this method, called *Greedy Entropy Search*, the next θ_{k+1} is chosen as the point that contributes the most to the entropy of P, i.e.,

$$
\theta_{k+1} = \arg\max_{\theta \in \Theta} -P(\theta) \log\big(P(\theta)\big).
$$

This selection criterion is greedy because it does not anticipate how the output of the simulation using θ_{k+1} would affect the entropy of P. Nevertheless, this criterion selects the point that is causing the entropy of P to be high. That is, a point θ_{k+1} with a good chance $P(\theta_{k+1})$ of being the real model, but with a high uncertainty $P(\theta_{k+1}) \log\big(\frac{1}{P(\theta_{k+1})}\big)$.

Random Embedding for Model Identification in the High Dimensional Space

For problems where the space Θ of physical properties has a high dimension D, the method presented in Algorithm 1 is not practical because the number of elements in discretized Θ is exponential in dimension D. This is a common problem in global search methods (Wang et al., 2016). In fact, it has been shown that Bayesian optimization techniques do not perform better than a random search when the dimension of the search space is too large (10 dimension in the experiment in (Ahmed, Shahriari, and Schmidt, 2016)). Therefore, Algorithm 1 cannot be directly used for robotic platforms with a large number of joints and parameters, such as the Tensegrity robot or compliant dexterous hands.

Dimensionality reduction is a popular solution to the problem of searching in high-dimensional spaces. This solution is particularly appealing in the context of this work because we are more interested in the accuracy of the predicted trajectory than in identifying the true underlying physical parameters. Mechanical models of motion tie together several parameters of an object. For example, in Coulomb's model, the mass and the friction of an object are used in a linear function to predict the motion of a sliding planar object. Therefore, one can map linearly these two parameters to a single parameter and still make accurate predictions of the motion.

Random embedding is an efficient and effective dimensionality reduction technique (Wang et al., 2016). Given a space of parameters Θ with dimension D, we generate a random matrix $A \in \mathscr{R}^{D \times d}$ that projects points from $\Theta \subset \mathbb{R}^D$ to a lower-dimensional space of parameters $\Omega \subset \mathbb{R}^d$ where $d < D$. Instead of discretizing Θ, we discretize Ω into a regular grid and map each point $\omega \in \Omega$ to a point θ in the

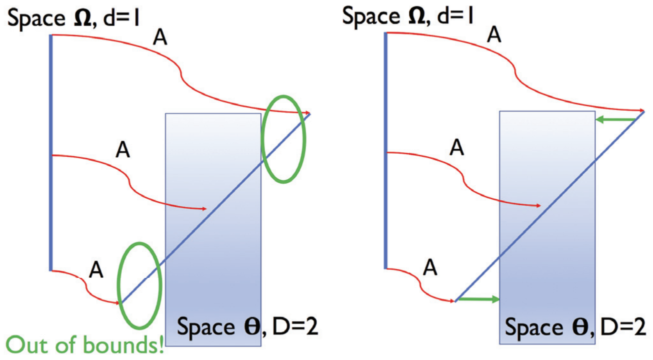


Figure 2: LEFT: A example of 1D-to-2D projection resulting in points outside the original domain. RIGHT: REMBO approaches this issue by projecting the point outside Θ to the nearest boundary point of Θ.

original high-dimensional space by using A, i.e. $\theta = A\omega$. One can show (Wang et al., 2016) that with probability one, $\min_{\theta\in\Theta} E(\theta) = \min_{\omega\in\Omega} E(A\omega)$ where E is the error function in Equation 1. Consequently, we run Algorithm 1 using discretized Ω as input instead of Θ. We project back the low-dimensional vectors $\omega \in \Omega$ to original parameter space Θ using $\theta = A\omega$ when we need to run the physical simulation to get the trajectory under a sampled value of ω.

However, For a randomly generated matrix A and point $\omega \in \Omega$, the corresponding high-dimensional vector $\theta = A\omega$ is not guaranteed to belong to Θ, but could instead lie anywhere within $\mathbb{R}^D$. The simulator may consider θ as invalid if it is outside of Θ as shown in Fig.2. Moreover, just doing a rejection sampling does not always work because most of the points could be rejected for being invalid in some cases. *Random EMbedding Bayesian Optimization (REMBO) (Wang et al., 2016)* addressed this issue simply by projecting the point outside Θ to the nearest boundary point of Θ.

Variational Auto Encoder for Model Identification in the High Dimensional Space

An auto encoder is a neural network that learns to reconstruct the input by going through a latent space, which is in a lower dimensional space than the original input space(Vincent et al., 2010). It has shown to be very useful in unsupervised learning of low dimensional representations. A variational auto encoder (VAE) adds an additional constraint that the latent space follows a prior distribution, usually assumed to be Gaussian (Kingma and Welling, 2014). This additional constraint makes the model more useful as a generative model, as it also learns to generate output from the prior distribution in addition to reconstruction.

We adapt the VAE and combine it with the Bayesian optimization process, as shown in Fig. 3. Firstly, the VAE is trained with randomly sampled physical parameter data θ to learn a low dimension embedding α. Once the VAE is optimized, the decoder part is used to project the low dimensional α back to the original physical parameter space θ. Thus, the Bayesian optimization process as detailed in Algorithm 1 can be done efficiently in the low dimensional space. The decoder can be seen as a learned non-linear version of the projection matrix A in REMBO.

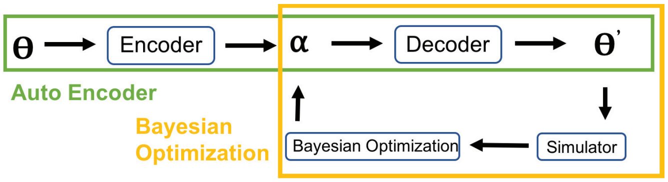


Figure 3: The auto encoder is trained first to learn the latent low dimensional embedding. Then Bayesian optimization is performed in this low dimensional space to search for the optimal parameter. The decoder is used to reconstruct the original 15 dimensional parameter in order to perform physical simulation.

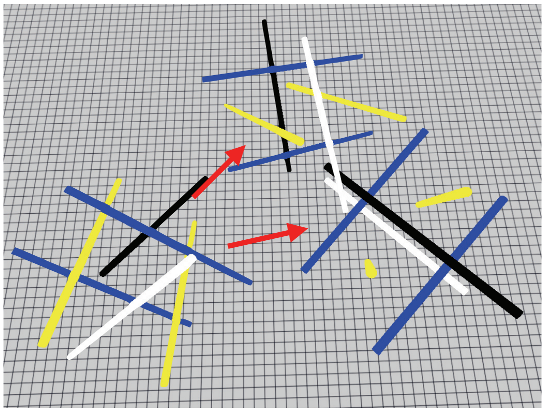


Figure 4: Simulation of the Tensegrity robot resulting in different states when executing the same control for different parameters.

Experimental Results

Setup: This experiment aims to identify the 15 parameters of the T6 model of the Tensegrity SuperBall robot in NASA's Tensegrity Robotics Toolkit (NTRT). The complex dynamics and high dimensionality of the robot make this problem very hard. Fig. 4 shows an example of the different results of applying the same control to the robot with 1% difference in the rod length (one of the 15 parameters). In absence of access to the real robot, the default values of the T6 model in NTRT are used as ground-truth. The Guided Policy Search (GPS) algorithm (Levine and Abbeel, 2014) was used to discover fast trajectories of several flops through iterative exploration and refinement (GPS controller).

The Greedy Entropy Search (GES) method is compared against random search, where random values of the parameters are selected within the ±10% range. Nevertheless, it is well-known that Bayesian optimization in high dimensions is difficult due to the exponential growth of the search space. To deal with this issue, the two dimensionality reduction methods, REMBO and VAE are used to reduce the dimensionality of the parameter space from 15 to 5.

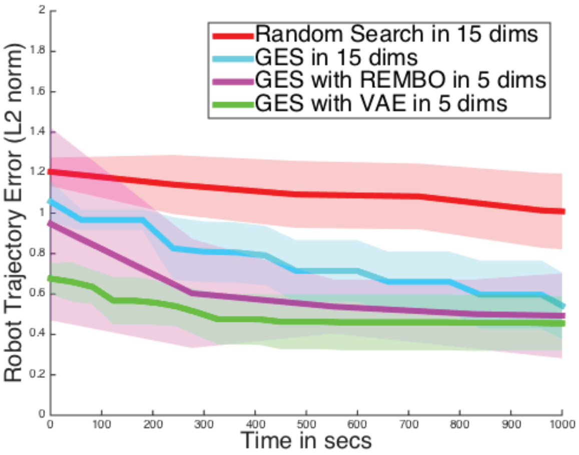


Figure 5: Test trajectory errors of different methods for the Tensegrity robot as a function of time budget for the parameter optimization process. Greedy Entropy Search in the 5-dimensional space using VAE achieves the lowest trajectory error, outperforming random search and Greedy Entropy Search in the original 15 dimensional space, as well as Greedy Entropy Search in the 5-dimensional space using REMBO.

The encoder and decoder of the VAE used in the experiment are both two-layer neural networks. The input dimension of the encoder and the output dimension of the decoder is 15, which is the dimension of the parameter space. The latent space is 5 dimensional. Between them is one layer of 400 dimensions. This dimension is chosen through cross validation by balancing accuracy and network complexity. The prior distribution of the latent space in the VAE is assumed to be $N(0, 1)$. Based on the three-sigma rule, when sampling between $[-3, 3]$, this interval should cover 99.7% of the latent space when the VAE is optimized. For REMBO, each time a random projection matrix is generated to project the parameters into $[0, 1]$.

To train the VAE, 10,000 training trajectories are generated. These trajectories are generated by running the GPS controller in the simulator with different physical parameters and adding random noise of up to ±10% to the default parameter values. This means each trajectory is generated under slightly different physical parameters.

Results: Fig. 5 shows the average error between the trajectories using the model parameters identified by different methods and the trajectories generated from the ground-truth simulator. When optimizing in the original 15-dim. space, as a data-efficient global optimization method, Bayesian optimization with Greedy Entropy Search outperformed random search. Further improvements are achieved by dimensionality reduction, making the search more efficient. Greedy Entropy Search in the 5-dimensional space using VAE achieves the lowest trajectory error, outperforming the method using REMBO. This shows that a learned better latent embedding enables more efficient parameter search in the Bayesian optimization process. A video showing examples of the Tensegrity robot locomotion can be found on https://youtu.be/lD31s0c_tqM.

Fig. 6 provides the errors for each of the parameter as a function of time budget for the parameter optimization process. Only the combination of Greedy Entropy Search with VAE achieves close to 1% error for all parameters. Some parameters may have stronger influence on the robot dynamics. An intelligent way to identify these parameters would be helpful to reduce the dimensionality of the parameter space and could be more informative than random embeddings. This will be a direction for future work.

Conclusion

This work proposes an information and data efficient framework for identifying physical parameters critical for robotic tasks, such as compliant robot locomotion. The framework aims to minimize the error between trajectories observed in experiments and those generated by a physics engine. To minimize the number of needed experiments, a Greedy variant of Entropy Search is proposed, which is shown to be data efficient. To solve high-dimensional challenges, this work integrates Greedy Entropy Search with a projection to a lower-dimensional space through random embedding or learning a latent embedding utilizing variational auto encoder. The evaluation of the proposed method against alternatives is favorable both in terms of identifying parameters more efficiently, as well as resulting in more accurate locomotion trajectories.

An interesting extension of this work would involve the identification of controls during the learning process that help in quickly minimizing the error. This can be a robust control process, which takes advantage of Bayesian Optimization's output in terms of a belief distribution for the identified parameters, so as to minimize entropy and maximize the safety of the experimentation process. Furthermore, it is interesting to compare the generality of the learned models and resulting control schemes that utilize them against completely model-free and end-to-end approaches for reinforcement learning and control.

References

Ahmed, M.; Shahriari, B.; and Schmidt, M. 2016. Do we need ”harmless” bayesian optimization and ”first-order” bayesian optimization? In *NIPS BayesOPT Workshop*.

Antonova, R.; Rai, A.; and Atkeson, C. G. 2016. Sample efficient optimization for learning controllers for bipedal locomotion. In *Humanoid Robots (Humanoids), 2016 IEEE-RAS 16th International Conference on*, 22–28. IEEE.

Bertsekas, D. P., and Tsitsiklis, J. N. 1996. *Neuro-Dynamic Programming*. Athena Scientific, 1st edition.

Bullet physics engine. [Online]. Available: www.bulletphysics.org.

Calandra, R.; Seyfarth, A.; Peters, J.; and Deisenroth, M. P. 2016. Bayesian optimization for learning gaits under uncertainty. *Annals of Mathematics and Artificial Intelligence (AMAI)* 76(1):5–23.

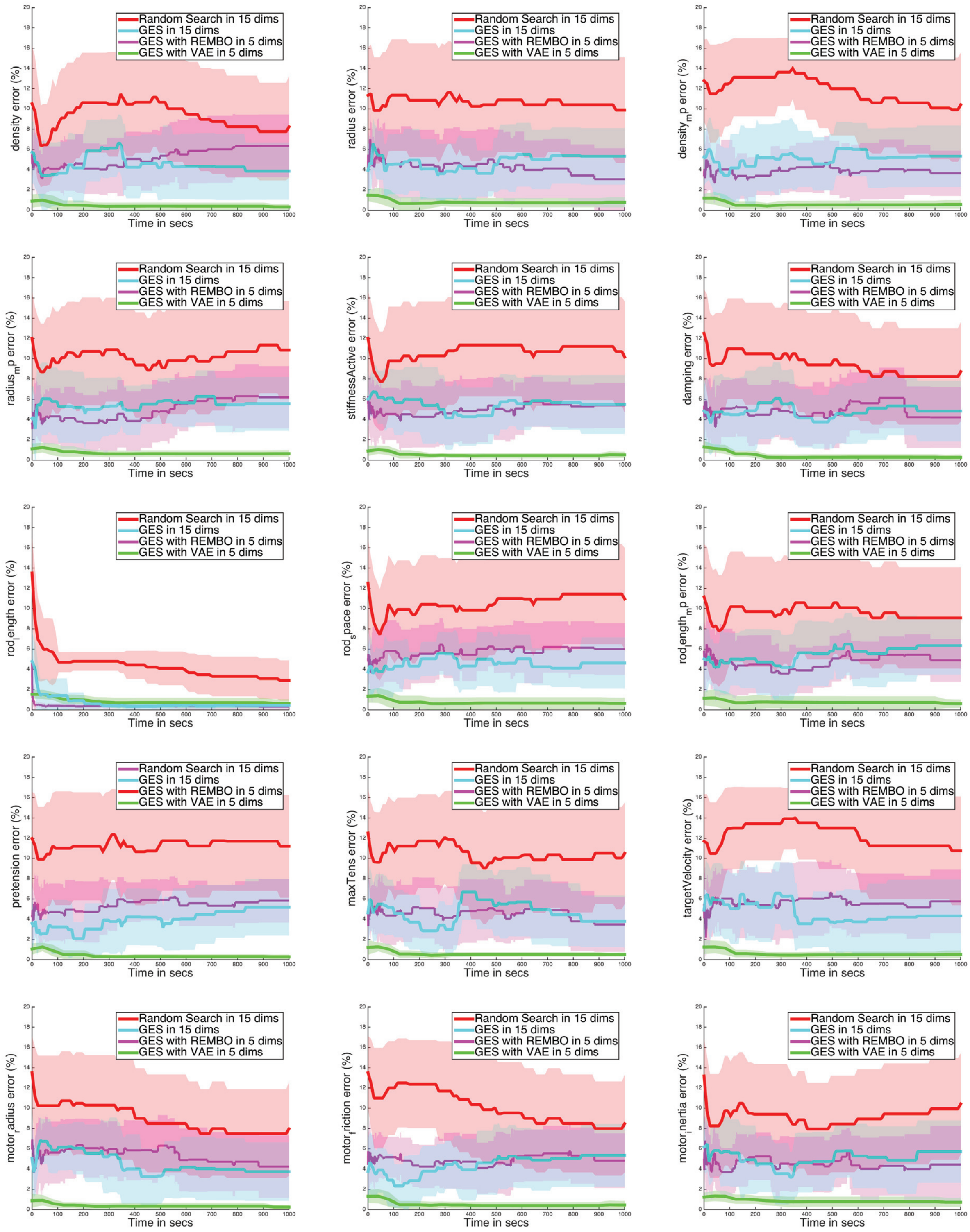


Figure 6: Each of the fifteen parameter error functions for the Tensegrity robot as a function of time budget for the parameter optimization process Greedy Entropy Search in the 5-dimensional space using VAE achieves the lowest error, which is less than 1% for all dimensions.

Caluwaerts, K.; Despraz, J.; Iscen, A.; Sabelhaus, A.; Bruce, J.; Schrauwen, B.; and SunSpiral, V. 2014. Design and control of compliant tensegrity robots through simulation and hardware validation. *Journal of The Royal Society Interface* 11(98).

DART physics egnine. [Online]. Available: http://dartsim.github.io.

Deisenroth, M.; Rasmussen, C.; and Fox, D. 2011. Learning to Control a Low-Cost Manipulator using Data-Efficient Reinforcement Learning. In *Robotics: Science and Systems (RSS)*.

Dogar, M.; Hsiao, K.; Ciocarlie, M.; and Srinivasa, S. 2012. Physics-Based Grasp Planning Through Clutter. In *Robotics: Science and Systems VIII*.

Erez, T.; Tassa, Y.; and Todorov, E. 2015. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ODE and physx. In *IEEE International Conference on Robotics and Automation, ICRA*, 4397–4404.

Geng, X.; Zhang, M.; Bruce, J.; Caluwaerts, K.; Vespignani, M.; SunSpiral, V.; Abbeel, P.; and Levine, S. 2016. Deep reinforcement learning for tensegrity robot locomotion. *CoRR* abs/1609.09049.

Havok physics engine. [Online]. Available: www.havok.com.

Hennig, P., and Schuler, C. J. 2012. Entropy Search for Information-Efficient Global Optimization. *Journal of Machine Learning Research* 13:1809–1837.

Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. In *ICLR*.

Kober, J., and Peters, J. R. 2009. Policy search for motor primitives in robotics. In *Advances in neural information processing systems*, 849–856.

Kober, J.; Bagnell, J. A. D.; and Peters, J. 2013. Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*.

Levine, S., and Abbeel, P. 2014. Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems (NIPS)*.

Lynch, K. M., and Mason, M. T. 1996. Stable pushing: Mechanics, control- lability, and planning. *IJRR* 18.

Marco, A.; Berkenkamp, F.; Hennig, P.; Schoellig, A. P.; Krause, A.; Schaal, S.; and Trimpe, S. 2017. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with bayesian optimization. In *2017 IEEE International Conference on Robotics and Automation, ICRA 2017, Singapore, Singapore, May 29 - June 3, 2017*, 1557–1563.

Merili, T.; Veloso, M.; and Akin, H. 2014. Push-manipulation of Complex Passive Mobile Objects Using Experimentally Acquired Motion Models. *Autonomous Robots* 1–13.

Mirletz, B. T.; Park, I.-W.; Quinn, R. D.; and SunSpiral, V. 2015. Towards bridging the reality gap between tensegrity simulation and robotic hardware. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; Petersen, S.; Beattie, C.; Sadik, A.; Antonoglou, I.; King, H.; Kumaran, D.; Wierstra, D.; Legg, S.; and Hassabis, D. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

MuJoCo physics engine. [Online]. Available: www.mujoco.org.

NTRT. NASA tensegrity robotics toolkit (NTRT). https://ti.arc.nasa.gov/tech/asr/intelligent-robotics/tensegrity/NTRT/.

Peters, J.; Mülling, K.; and Altün, Y. 2010. Relative entropy policy search. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI 2010)*, 1607–1612.

PhysX physics engine. [Online]. Available: www.geforce.com/hardware/technology/physx.

Rasmussen, C. E., and Williams, C. K. I. 2005. *Gaussian Processes for Machine Learning*. The MIT Press.

Scholz, J.; Levihn, M.; Isbell, C. L.; and Wingate, D. 2014. A Physics-Based Model Prior for Object-Oriented MDPs. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*.

Shahriari, B.; Swersky, K.; Wang, Z.; Adams, R. P.; and de Freitas, N. 2016. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE* 104(1):148–175.

Sutton, R. S., and Barto, A. G. 1998. *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st edition.

Vincent, P.; Larochelle, H.; Lajoie, I.; Bengio, Y.; and Manzagol, P.-A. 2010. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research* 11(Dec):3371–3408.

Wang, Z.; Hutter, F.; Zoghi, M.; Matheson, D.; and de Feitas, N. 2016. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research* 55:361–387.

Zhou, J.; Paolini, R.; Bagnell, J. A.; and Mason, M. T. 2016. A convex polynomial force-motion model for planar sliding: Identification and application. In *2016 IEEE International Conference on Robotics and Automation, ICRA 2016, Stockholm, Sweden, May 16-21, 2016*, 372–377.

Learning, Inference, and Control of Multi-Agent Systems

Learning Machines

Magnus Boman, Magnus Sahlgren, Olof Görnerup, Daniel Gillblad
Swedish Institute of Computer Science, RISE SICS
Box 1263, SE-164 29 Kista, Sweden

Abstract

This position paper explicates the notion of learning machines and how they may cooperate and compete to scale over multiple domains. We argue that important problem applications very soon will start to benefit from cross-domain learning housed in learning machines. We outline an architecture involving human-machine interplay, including education of, and assessments of the value of learning machines.

Introduction

This position paper has been written with the intent to spark interest in discussing how to best design multiple-machine systems that learn for applications important to humans or to the planet. We thus consider a class of problems much too hard for adequate solutions to be completed in the next decade. Because elements of such designs are fully understandable already today, we argue that a pro-active discussion could help focus research and development efforts on providing value to humans and their environment.

A *learning machine* (LM) is an autonomous self-regulating open reasoning machine that actively learns in a decentralized manner, over multiple domains. The autonomy of the machine allows it to move between domains, situated, between abstract domain models, or both. Feedback allows it to self-correct its models and its processes of education. *Openness* pertains to simple I/O as well as to the machine changing its component materials. The I/O behavior regulates human-machine- but also machine-machine communication. That its learning is *active* means that it can pursue learning goals in batch, and reinforce by means of self-testing and evolutionary learning. The process of education may be represented in the learning machine as a classical planning-inference-action loop, but it may also be more exploratory and, e.g., replace planning by serendipitous stimulus learning. We prefer the modernized loop of perception-reasoning-interaction (Figure 1). Its *decentralized* nature lets it interact, but also links to its autonomous migration in that a machine may create copies of itself, or otherwise modify its appearance, even if no human or machine observes these changes. The models employed inside the machine are essentially statistical learning theory models, although the knowledge representation may vary, for the purposes of experimentation or for domain tailoring. Finally, the *multiple domain* aspect goes beyond transfer learning, in that the knowledge representation lets the machine learn where it is and what is appropriate in this particular setting, situated or not. The latter point includes ethics, norms, and other behavioral constraints, all subject to dynamic control modulo the autonomy of the machine.

Education of LMs may be the result of human-LM interaction, but it may also result from LM2LM interaction. Given the current emphasis in machine learning research on reinforcement learning and its high performance in limited application domains, we would like to see more efforts towards adoption and delegation mechanisms in multi-LM systems, and for important applications. Our starting point is Internet-based psychiatry. Cross-domain use of such an LM may be used in different patient groups, varying, e.g., age intervals, geographic location, or syndrome. In a national project in Sweden, we are currently building LMs for 16+ age patients with social anxiety, social phobia, or depression, in cooperation with Karolinska Institute, at Stockholm. While our short-term goals are finding predictor sets for patient adherence and activity (Yardley et al. 2016), the long-term goals are to define and implement multi-LM systems for settings in which multiple humans interact, under (relatively forgiving, compared to, e.g., the OpenAI team tackling DOTA2 5v5 in the future) real-time constraints. These are thus multi-LM systems in which humans must be modeled, and in which LMs must adapt accordingly, in real-time. A possible future scenario is a discussion in natural language between a group of human and artificial therapists, taking place in an Internet psychiatry room (in which there are never any patients).

Education

The education of learning machines in the sense conceived by Turing is now possible at scale, thanks to adequate computational resources, enough data, and the modularity of LM capabilities (allowing for compartmentalized training and a reductionist approach to machine education). We are interested in both hardware and software self-modification, i.e. what Turing called *screwdriver* and *paper* interference, respectively. In his terminology, we take on the role of LM

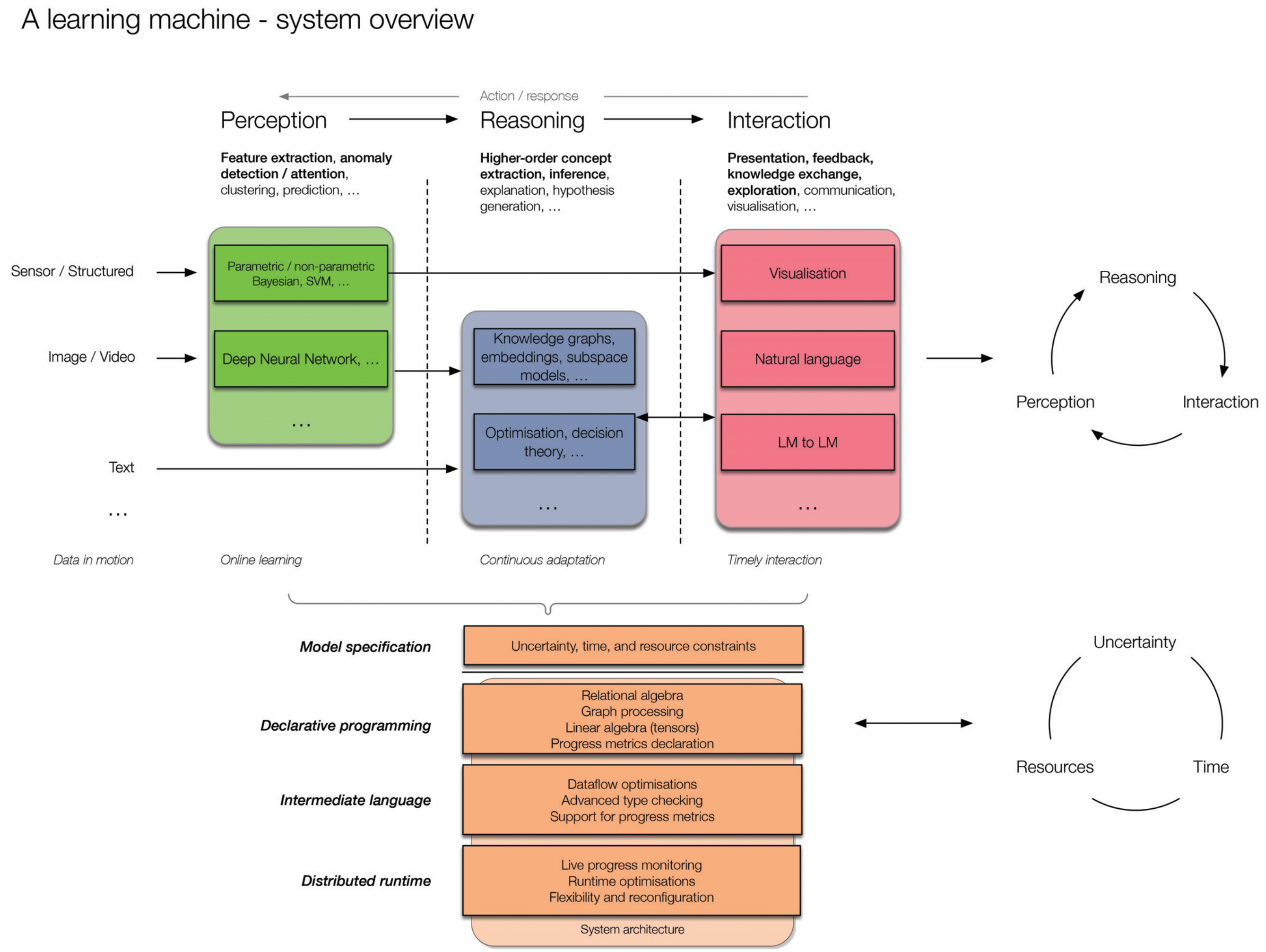

A learning machine - system overview
Action / response
Perception
Reasoning
Interaction
Feature extraction, anomaly detection / attention, clustering, prediction, …
Higher-order concept extraction, inference, explanation, hypothesis generation, …
Presentation, feedback, knowledge exchange, exploration, communication, visualisation, …
Sensor / Structured
Image / Video
Text
…
Parametric / non-parametric Bayesian, SVM, …
Deep Neural Network, …
…
Knowledge graphs, embeddings, subspace models, …
Optimisation, decision theory, …
…
Visualisation
Natural language
LM to LM
…
Data in motion
Online learning
Continuous adaptation
Timely interaction
Reasoning
Perception
Interaction
Model specification
Uncertainty, time, and resource constraints
Declarative programming
Relational algebra
Graph processing
Linear algebra (tensors)
Progress metrics declaration
Intermediate language
Dataflow optimisations
Advanced type checking
Support for progress metrics
Distributed runtime
Live progress monitoring
Runtime optimisations
Flexibility and reconfiguration
System architecture
Uncertainty
Resources
Time

Figure 1: We conceptually divide an LM system into perception, reasoning, and interaction, possibly closed in an iterative loop where an action performed by the LM affects the environment as well as input data. As approaches and algorithms may overlap, and as data types and their processing vary, the figure should be seen as an illustrative example for structuring reusable components. The perception part of the system largely performs supervised or unsupervised machine learning tasks such as classification, prediction, clustering, and anomaly detection, the output of which can be used directly for interaction, or as input to LM reasoning. In the latter case, the task performed by the perception layer can be viewed as lifting the level of abstraction from raw input data to more abstract concepts. An example would be the use of the units of the last few layers of a deep convolutional neural network as a representation of discrete abstract features in image data. The reasoning system involves making optimal decisions under uncertainty, planning, generating hypotheses, explaining hypotheses and detailing its reasoning, and responding to queries. A central challenge is to find general representations of knowledge that can continuously learn arbitrary relations from data. This is particularly challenging when meta-level reasoning is involved, which is necessary for cross-domain applications. The interaction part may involve, but is not restricted to, visualization and natural language for human interaction or direct LM2LM interaction. We envision that LM algorithms rest on a system architecture, a substrate for LMs, that can manage the trade-off between uncertainty or precision, time, and computing resources, and that can operate on very large data sets and streaming data. To enable end-to-end optimization of a whole LM, we need declarative programming models that encapsulate linear (or tensor) algebra, graph processing, and relational algebra. This would be translated to an intermediate language performing dataflow optimizations and type checking, running on distributed environments that allow for performing run-time optimization for multiple architectural choices and criteria.

educators, and as “highly competent schoolmasters” we are denied detailed information on the inner workings of the machine; as designers we are not denied this, but as “mechanics” we do not educate the LM; and “they would be able to converse with each other to sharpen their wits” (Turing 1948). Having pattern recognition methods and classification of new non-trivial features built into an LM makes inductive inference efficient and possible to validate, even in cases of unsupervised education. We also subscribe to the original idea of an LM as a guessworks expert, having the ability of making correct assignments or statements, *as if* the machine had guessed a rule (Gamba 1961), even though the LM were presented with no rules during learning. Probabilistic reasoning based on weight adjustments and discrete success/failure-based learning turns LM education into a branch of probability theory, leaning in particular towards statistical learning theory (Vapnik 1995). In general, success-reinforced models are averaging models, and an understanding of higher-order concepts on the LM’s part and how these related to assumptions of underlying distributions are of great importance, not least for understanding bias and risks associated with over-learning. The education material is in all interesting cases, as in our Internet psychiatry case, multi-modal: we may incorporate text, images, biobank data, as well as random noise of various forms. A particularly challenging problem is how to learn how to learn, e.g., how to train a neural network to dynamically adjust models of perfect precision/recall (Reed and de Freitas 2015), or if the interest is towards mimicking biological systems, Universal LMs (Duch and Maszczyk 2009).

Language learning

A successful application domain for AI in recent years has been to learn language representations that can be used to quantify semantic similarity between linguistic items. Such models – normally referred to as *word embeddings* – are data-intensive statistical algorithms (often implemented as neural networks) that learn to generalize word usage patterns from observed samples of language use. Training such a model is currently done by throwing as much data as possible to it, and parameterizing by informed guessworks. Each language model is thus employed as part of the learning process by observing as much as possible of human language use. In a weak sense, this is like a child trying to learn language by only *listening* to her parents; it might go a long way, but it is essentially underdetermined; only through feedback from *use* and *interaction* that the child becomes a proficient language user.

We suggest that the same should apply to a learning machine applied to language data. In analogy with a human language learner, listening to, and interacting with, a teacher (which in the case of the LM is a human, and in the case of the child can be a parent or a school teacher) provides strong and (often) precise feedback. However, interacting with other language learners (which in the case of an LM is other LMs, and in the case of a child is other children) may provide more frequent, but less precise feedback. We are particularly interested in LM2LM interaction in this respect; can LMs learn language from each other and not only from observations of human linguistic behavior? Would such multi-LM systems acquire their own language, and how do you capture dialogue dynamics? Recurrence, convolution, higher-order relations, and more generally the concept of hidden variables and layers are part of the current answer, but the problem is understudied (Jordan 1989) (Elman 1990).

Perception and Reasoning

LMs learn to extract relevant information from sensory inputs and reason about this information using internal representations such as knowledge graphs. Symbolic reasoning is then supported by higher-order concept discovery—an ability to abstract concepts into other concepts—hypothesis generation, and inference. A key feature then is adaptability: representations are plastic and knowledge is always questioned as an LM may arguably be designed to be eager to learn as well as curious.

LMs can collectively perceive, learn, and reason across domains, resulting in improved learning rates and versatility, where both sensory inputs and gained knowledge are shared among machines. Cooperating LMs, sharing experiences, lessons learned from pitfalls, etc., also increase robustness and adaptability since a collective of LMs is more resilient than a single, possibly myopic or even solipsistic machine. The boundary between LMs may then be obscured, where a system of LMs can be interpreted as a meta-machine that could constitute an LM in its own right.

Treating LMs as a collective comes with multiple systemic challenges pertaining to self-organizing dynamics. How do we for example avoid negative spirals causes by feedback loops in the system as to ensure robust collective dynamics? Collective learning also requires both robust and flexible LM2LM communication and knowledge transfer, and commonly agreed knowledge representations. Here, delegation/adoption loops constitute one alternative (Castelfranchi and Falcone 1998), and so do norm-regulated systems (Boman 1999) and lower-level representations, like contract nets (Smith 1977). These representations can themselves then be subject to learning among machines. An LM needs to be empathic, at least in a weak sense, understanding experiences and internal states of other LMs.

Conclusion

Reinforcement learning in toy domains, such as turn-taking board games or early arcade-style computer games, constitutes essentially a structuralist representation of knowledge. Because the game representation is in full grasped by the learner in terms of pixels or board piece configurations—what is learned is a surface structure model—making generalizations difficult. That the results are still impressive is in part due to how the results are presented by the educators of the learner. It is easy then easy to forget that when we consider the future of learning machines, with reinforcement learning as one candidate for how to educate them, these educators and their designs are not part of the model. In other words, reinforcement learners do not, to paraphrase Turing, roam the countryside and learn from their experience. By contrast, our interest lies in the pursuit of knowledge sitting in learning machines, and how to best support it by design. In this position paper, we have outlined an architecture making such behavior possible, and we ourselves seek to complete implementations in line with this architecture in the next few years in cross-domain fashion. Our first field of application is Internet psychiatry, but we will complete implementations in parallel in several other domains in order to pinpoint exactly how cross-domain learning machines may provide value to humans. Our work is open-ended and collaborative and we are currently establishing academic as well as industry networks of cooperation to make this possible.

References

Boman, M. 1999. Norms in artificial decision making. *Artif. Intell. Law* 7(1):17–35.

Castelfranchi, C., and Falcone, R. 1998. Towards a theory of delegation for agent-based systems. *Robotics and Autonomous Systems* 24(3):141 – 157. Multi-Agent Rationality.

Duch, W., and Maszczyk, T. 2009. Universal learning machines. In Leung, C. S.; Lee, M.; and Chan, J. H., eds., *Neural Information Processing*, 206–215. Berlin, Heidelberg: Springer Berlin Heidelberg.

Elman, J. L. 1990. Finding structure in time. *Cognitive Science* 14(2):179 – 211.

Gamba, A. 1961. Optimum performance of learning machines. *Proceedings of the IRE* 49:349 – 350.

Jordan, M. I. 1989. Serial order: A parallel, distributed processing approach. In Elman, J. L., and Rumelhart, D. E., eds., *Advances in Connectionist Theory: Speech*. Erlbaum.

Reed, S. E., and de Freitas, N. 2015. Neural programmer-interpreters. *CoRR* abs/1511.06279.

Smith, R. G. 1977. The contract net: A formalism for the control of distributed problem solving. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'77, 472–472. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Turing, A. 1948. Intelligent machinery, A heretical theory. In Ince, D., ed., *Collected Works of A. M. Turing Volume 1: Mechanical Intelligence*. North Holland.

Vapnik, V. N. 1995. *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc.

Yardley, L.; Spring, B. J.; Riper, H.; Morrison, L. G.; Crane, D. H.; Curtis, K.; Merchant, G. C.; Naughton, F.; and Blandford, A. 2016. Understanding and promoting effective engagement with digital behavior change interventions. *American Journal of Preventive Medicine* 51(5):833 – 842.

Learning in Ad-Hoc Anti-Coordination Scenarios

Panayiotis Danassis, Boi Faltings
Artificial Intelligence Laboratory (LIA), École Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
Email: {panayiotis.danassis, boi.faltings}@epfl.ch

Abstract

We present a brief overview of learning dynamics for anti-coordination in ad-hoc scenarios. Specifically, we consider multi-armed bandit algorithms, reinforcement learning, and symmetric strategies for the repeated resource allocation game. In a multi-agent system with dynamic population where every agent is able to learn, the anti-coordination problem exhibits unique challenges. Thus, it is essential for the success of a joint plan that the agents can quickly and robustly learn their optimal behavior. In this work we will focus on convergence rate, efficiency, and fairness in the final outcome.

1 Introduction

In multi-agent systems, most scenarios require coordination on the same value which involves solving the consensus problem, a well-studied problem in distributed computing (Coulouris, Dollimore, and Kindberg 2005). However, there are also many situations where agents are required to choose distinct actions as in role allocation (e.g. teammates during a game), task assignment (e.g. employees of a factory), resource allocation (e.g. wireless bandwidth (channels) for IoT devices, parking spaces and/or charging stations for autonomous vehicles) etc. This is called *anti-coordination*. Figure 1 provides an illustrative example. For simplicity, we focus on resource allocation scenarios, although the considered learning models can be applied in any analogous anti-coordination scenario.

Anti-coordination in multi-agent systems presents many unique challenges. First, it requires agents to take different actions while facing the same problem. Hence, we need agents that are able to learn to behave *differently* in the presence of (possibly) identical agents while having similar preferences across their available actions. An autonomous vehicle would prefer the route with the least traffic, an IoT device would prefer the higher bandwidth channel, a bidding agent participating in multiple auctions would prefer the one with the fewer participants, etc. Nevertheless, in order to achieve high efficiency, we need some agents to take less desirable actions. An added challenge is ensuring fairness in the final outcome, i.e. make sure that those agents are not exploited, and ensuring that self-interested, rational agents are not able to manipulate the algorithm to maximize their utility. Furthermore, in real world applications agents tend to receive only partial feedback; i.e. each agent is only aware of his own history of action/reward pairs. Hence, we require completely uncoupled learning rules and agents that are capable of achieving high efficiency and fast convergence in such information-restrictive settings. Finally, intra-agent interactions might need to take place in an ad-hoc fashion, which brings forth the need for robust agents that are able to coordinate with previously un-encountered participants (Stone et al. 2010). However, planning in such environments becomes even more challenging. Part of this difficulty stems from the lack of responsiveness and/or communication between the participants.

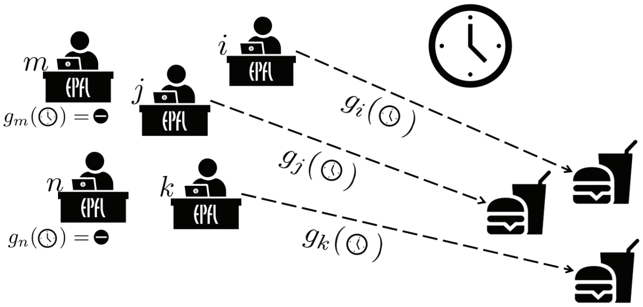


Figure 1: Every day N employees have lunch at a cafeteria which accommodates R patrons, thus their goal is to anti-coordinate their lunch breaks. Each of them has a strategy ($g_n, \forall n \in \{1, \dots, N\}$) for selecting their lunch time. All employees have similar preferences (e.g. have lunch between 12p.m. - 2p.m., find an empty seat etc.). Each time they attempt to have a lunch break, they update their strategy based on their personal feedback of success or failure.

Little work has been done in anti-coordination problems as compared to classical coordination scenarios. Mapping anti-coordination to the consensus problem results in an exponential expansion of the solution space. Hence, special effort is required from a learning perspective. In this paper we present a brief comparative overview of multi-agent learning paradigms applicable to the anti-coordination setting. The rest of the paper is organized as follows. Section

2 provide a formal definition of the repeated resource allocation (anti-coordination) problem, Section 3 presents the evaluated multi-agent learning models, and finally, Section 4 concludes the paper.

2 Preliminaries

2.1 The Repeated Resource Allocation Problem

In this section we formally define the repeated resource allocation problem. The goal for the agents is to maximize their discounted cumulative payoff. We refer to a 'resource' as any element that can be successfully assigned to only one agent at a time. At each time-step, $\mathcal{N} = \{1, \dots, N\}$ agents try to access $\mathcal{R} = \{1, \dots, R\}$ identical and indivisible resources. The set of available actions is denoted as $\mathcal{A} = \{Y, A_1, \dots, A_R\}$, where Y refers to yielding, while A_r refers to accessing resource r. We assume that access to a resource is slotted and of equal duration. A successful access yields a positive payoff, while no access has a payoff of 0. If more than one agent accesses a resource simultaneously, a collision occurs and the colliding parties incur a cost $\zeta < 0$. The payoff function is defined by Equation 1, where a_n denotes agent n's action, and $a_{-n} = \times_{\forall n' \in \mathcal{N} \setminus \{n\}} a_{n'}$ the joint action for the rest of the agents.

$$u_n(a_n, a_{-n}) = \begin{cases} 0, & \text{if } a_n = Y \\ 1, & \text{if } a_n \neq Y \wedge a_i \neq a_n, \forall i \neq n \\ \zeta, & \text{otherwise} \end{cases} \quad (1)$$

In accordance to real-world phenomena we furthermore assume that the agents receive only partial feedback of success or failure; i.e. each agent n is only aware of his own history of action/reward pairs, $\mathcal{H}_n^t = \{(\alpha_n^\tau, u_n(\alpha_n^\tau, \alpha_{-n}^\tau))_{\forall \tau \leq t}\}$. The payoff matrix of the stage-game of a simple 1-resource, 2-agents, repeated resource allocation game is presented in Figure 2.

Finally, we assume that the agents can observe side information (context) from their environment at each time-step t (e.g. time, date etc. in the example of Figure 1), before taking their action. Let $\mathcal{K} = \{1, \dots, K\}$ denote the context space. We do not assume any a priori relation between the context space and the problem. The only constraint is that the context values should repeat periodically. In this work we assume that the context is a set of random integers. The motivation behind the introduction of the context space will become apparent in the following section. In short, we want to achieve high efficiency and fairness. In anti-coordination games with completely uncoupled learning rules such a goal is hard to attain since the aforesaid rules do not allow for correlation between the agents. The introduction of a common signal (such as the proposed context) resolves that issue.

2.2 Solution Concepts

In this section we examine possible game theory[1] solution concepts of the repeated resource allocation game, focusing on the following two axes:

	Y	A
Y	$0, 0$	$0, 1$
A	$1, 0$	ζ, ζ

Figure 2: Resource allocation game, $R = 1, N = 2$. Two agents want to access a single resource. Both of them have two actions, either to yield (Y), and get a payoff of 0, or access (A). If only one of the agents accesses the resource, he gets a payoff of 1. But if both of them access the resource at the same time, they collide and both incur a cost $\zeta < 0$.

i *Efficiency*: Percentage of utilized resources after convergence (alternatively, social welfare).

ii *Ex-post Fairness*: Equality of allotted resources after convergence (alternatively, ex-post expected payoff).

As a measure of fairness, we will use the Jain index (Jain, Chiu, and Hawe 1998). The Jain index exhibits a lot of desirable properties such as: population size independence, continuity, scale and metric independence, and boundedness. For a resource allocation game of N users, such that the n^{th} user receives an (expected) allocation of $w_n \geq 0$ resources, the Jain index is given by Equation 2. This equation measures the equality of allocation $\mathbf{w} = (w_1, \dots, w_N)^\top$. An allocation is considered fair, iff $\mathbb{J}(\mathbf{w}) = 1$.

$$\mathbb{J}(\mathbf{w}) = \frac{\left|\sum_{n \in \mathcal{N}} w_n\right|^2}{N \sum_{n \in \mathcal{N}} w_n^2} \quad (2)$$

Resource allocation games often admit undesirable equilibria; asymmetric pure Nash equilibria (PNE) which are efficient but not fair, or symmetric mixed-strategy Nash equilibria (MNE) which are fair but not efficient. For example, the set of asymmetric PNE corresponds to R agents accessing while $N - R$ yield. This results to 100% efficiency, but $\mathbb{J}_{PNE}(\mathbf{w}) = \frac{R^2}{NR} = \frac{R}{N}$. In the symmetric MNE, each agent decides to access with probability $Pr[\mathcal{A} \setminus \{Y\}] = \min\left\{R\left(1 - \sqrt[N-1]{\frac{|\zeta|}{1+|\zeta|}}\right), 1\right\}$ and then chooses which resource to access uniformly at random (Cigler 2013)). The latter results to expected $\mathbb{J}_{MNE}(\mathbf{w}) = 1$, but 0% expected efficiency (assuming small number of resources, R). As such, the aforementioned equilibria are rather undesirable. We can overcome the previously mentioned drawbacks using the notion of correlated equilibria (Aumann 1974).

Correlated equilibria (CE) are a superset of Nash equilibria. They allow for dependencies amongst the the agents' probability distributions, thus the optimization takes place on the joint action space. Correlated equilibria are desired solution concepts in resource allocation games, as they allow for efficient and fair solutions by avoiding positive probability mass on less desirable outcomes. Moreover, an optimal correlated equilibrium for resource allocation games may be found in polynomial time (Papadimitriou and Roughgarden 2008). Subsequently, a central coordinator who possesses complete information can recommend an action to each agent. Yet, an omniscient central coordinator is not always available, and in real-world applications with partial

[1] See (Nisan et al. 2007) for an introduction to game theory.

observability agents might not be willing to trust such recommendations. In a multi-agent scenario we are interested in agents who are able to *learn*; adapt their strategies and converge to an equilibrium. In order to be able to reach richer solution concepts, like correlated equilibria, the agents require a common signal upon which they can learn to anti-coordinate their actions. Hence the introduction of the environmental context, proposed in Section 2.1.

3 Overview of Learning Approaches

In this section we will outline potential multi-agent learning approaches for tackling the anti-coordination problem. We will examine bandit algorithms, reinforcement learning algorithms, and finally, symmetric equilibria for the repeated resource allocation game. We will focus on bimatrix (2-agents, 1-resource) games since, in spite of their simple form, they present many challenges in multi-agent learning scenarios (Littman and Stone 2002).

3.1 Ad-hoc Coordination & Multi-armed Bandit Algorithms

In ad-hoc multi-agent coordination the goal is to design autonomous agents that achieve high flexibility and efficiency in a setting that admits no prior coordination between the participants (Stone et al. 2010). Typical scenarios include the use of Monte Carlo algorithms (Barrett et al. 2017), Bayesian learning (Albrecht, Crandall, and Ramamoorthy 2016), or bandit algorithms (Chakraborty et al. 2017), (Barrett and Stone 2011). Traditionally, ad-hoc approaches suffer from slow learning, which makes ad-hoc coordination a very ambitious goal for real-life applications. Due to their ability to learn from partial feedback, bandit algorithms would be the natural choice for solving the anti-coordination problem in an ad-hoc setting.

In multi-armed bandit problems an agent is given a number of arms and at each time-step has to decide which arm to pull to get the maximum expected reward. Bandit (or no-regret) algorithms typically minimize the total regret of each agent, which is the difference between the expected received payoff and the payoff of the best strategy in hindsight. Additionally, they satisfy incentive constraints for rational agents since they constitute an approximate correlated or coarse correlated equilibrium (Nisan et al. 2007). Nevertheless, the studied problem presents many challenges: there is no stationary distribution (adversarial rewards), all agents are able to learn (similar to recursive modeling), and yielding gives a reward of 0 which might be a desirable option for minimizing regret, but not in respect to fairness.

To better understand these limitations, we evaluate three state-of-the-art, well established adversarial bandit algorithms, namely the EXP3 (Auer et al. 2002), the EXP4 (Auer et al. 2002), and the EXP4.P (Beygelzimer et al. 2011). The last two belong to a variant of multi-armed bandits, called contextual bandits[2], that is, at each time-step t, they can exploit the observed context $k_t \in \mathcal{K}$ before making their decision. As such, the chosen arm can be different depending on the context. Moreover, the EXP4.P combines the confidence bounds of UCB1 (Auer, Cesa-Bianchi, and Fischer 2002) with the EXP4 to achieve the same regret as EXP4 but with high probability. Figure 3 depicts the total utilization of resources for the 1-resource, 2-agents allocation game of Figure 2. The x-axis is in logarithmic scale, and the reported values are the average over 128 runs of the same simulation. The input parameters for the EXP family of algorithms are set to their optimal values, as prescribed in (Auer et al. 2002), and (Beygelzimer et al. 2011), assuming time horizon of $T = 10^5$ time-steps[3]. As depicted, all of the evaluated algorithms take a significant number of time-steps to reach a high utilization state, never achieve 100% efficiency, and exhibit high variance.

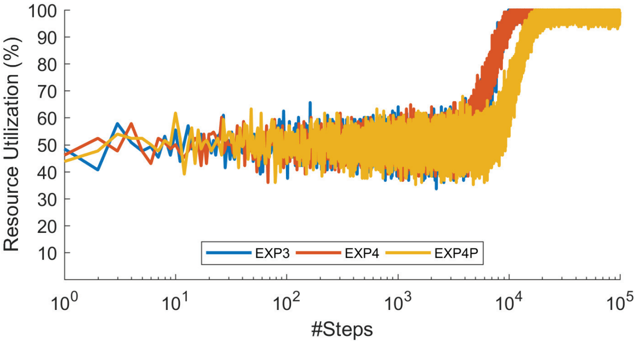


Figure 3: Resource utilization over time achieved by the employed bandit algorithms in the 1-resource, 2-agents allocation game of Figure 2 (x-axis in logarithmic scale).

Along with efficiency, we are interested in the fairness of the final outcome. Being able to achieve both is of the utmost importance for the adoption of such learning paradigms in real-world applications. The evaluated bandit algorithms exhibit considerably low fairness, specifically: $\mathbb{J}_{EXP3}(\mathbf{w}) = 0.50$, $\mathbb{J}_{EXP4}(\mathbf{w}) = 0.76$, $\mathbb{J}_{EXP4.P}(\mathbf{w}) = 0.73$. As a matter of fact, EXP3's achieved fairness is equal to that of an unfair asymmetric PNE: $\mathbb{J}_{PNE}(\mathbf{w}) = \frac{R}{N} = 0.5 = \mathbb{J}_{EXP3}(\mathbf{w})$. The contextual bandits performed somewhat better but, considering the simplicity of the evaluated example, not good enough. This leads to suggest that the evaluated contextual bandit algorithms are unable to handle the large policy space of anti-coordination games.

3.2 Reinforcement Learning & Replicator Dynamics

Closely related to the bandit algorithms of Section 3.1 is reinforcement learning. Reinforcement learning is based on the concept of learning through the interactions with the environment. An agent takes an action, observes some feedback from the environment, and updates his policy so as to maximize some notion of cumulative reward. The most eminent example of such an algorithm is Q-learning (Watkins

[2] See (Zhou 2015) for a survey on contextual bandits.

[3] Note the high sensitivity to the input parameter ($\gamma \in (0, 1]$), which is another crucial shortcoming of the studied bandit algorithms in ad-hoc scenarios.

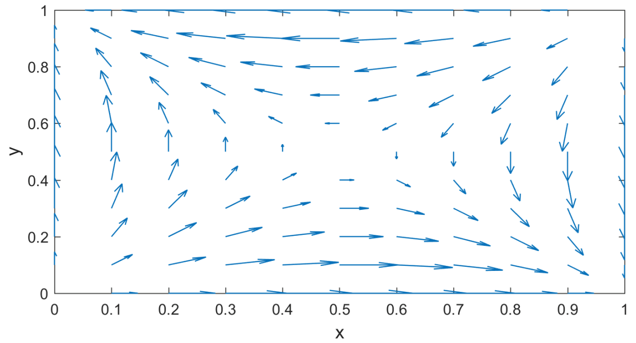


Figure 4: The replicator dynamics, plotted in the unit simplex, for the 1-resource, 2-agents allocation game of Figure 2. x denotes the first agent's probability of playing the first action (Y), while y denotes the second agent's probability of playing the first action (Y). The probabilities of playing the second action (A) are $1 - x$ and $1 - y$ respectively.

and Dayan 1992) which solves Bellman's optimality equation (Bellman 2013) using an iterative approximation procedure. A detailed taxonomy of multi-agent reinforcement learning algorithms can be found in (Busoniu, Babuska, and De Schutter 2008).

There is a formal relationship between reinforcement learning and the replicator dynamics of evolutionary game theory (Bloembergen et al. 2015), hence reinforcement learning algorithms can satisfy our incentive constraints. Evolutionary game theory (EGT)[4] differs from classical game theory in that it focuses on the dynamics of the learning process (strategy change). In a multi-agent system in which agents adapt their behavior in response to strategic interactions with other agents, evolutionary game theory provides a solid mechanism to analyze and understand it (Tuyls and Parsons 2007). Evolutionary game theory is built around the replicator equations:

$$\dot{x_i} = x_i \left[f_i(\mathbf{x}) - \phi(\mathbf{x}) \right] \tag{3}$$

Equation 3 describes the evolution of a population ($\mathbf{x}$) of individuals (x_i) over time, or alternatively (and more befitting to multi-agent learning), the evolution of an agent's strategy $\mathbf{x} = (x_1, \dots, x_R)^\top$. In the latter interpretation, the population share of each type ($x_i : 0 \leq x_i \leq 1, \forall i$) represents the probability of selecting action a_i, $f_i(\mathbf{x})$ is the fitness (utility) of action a_i, $\phi(\mathbf{x}) = \sum_j x_j f_j(\mathbf{x})$ is the weighted average fitness, and $\dot{x_i} = dx_i/dt$. For the the two agent game of Figure 2, we can rewrite Equation 3 for the strategy vector of the first agent $\mathbf{x}$ as:

$$\dot{x_i} = x_i \left[(U\mathbf{y})_i - \mathbf{x}^\top U \mathbf{y} \right] \tag{4}$$

where U is the payoff matrix (similar for $\mathbf{y}$).

Finding the optimal policy in a multi-agent system where all agents learn simultaneously is inherently more complex. Each agent is faced with a moving-target learning problem.

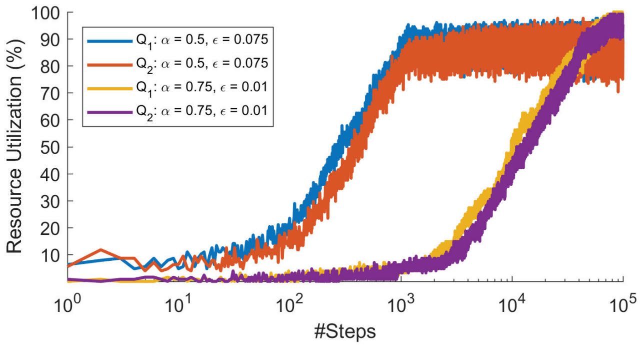


Figure 5: Resource utilization over time achieved by Q-learning in the 1-resource, 2-agents allocation game of Figure 2 (x-axis in logarithmic scale).

Changes in the policy of one agent can affect the rewards and thus have a cascading effect on the optimal policies of the others. Furthermore, just as with the bandit algorithms, the adaptation of such dynamics in real-world multi-agent problems requires fairness guarantees. An insight to the quality of the final allocation can be provided by examining the replicator dynamics (Equation 4) of the simple 1-resource, 2-agents allocation game of Figure 2, depicted in Figure 4. As seen by the plot, the two evolutionary stable strategies are the two unfair asymmetric PNE, (Y, A) and (A, Y). Moreover, Figure 5 depicts the total utilization of resources of two Q-learning approaches (the reported values are the average over 128 runs of the same simulation). The Q_1 approach uses the context as its state, while the Q_2 approach uses both the context and the former action as the state. The intuition behind Q_2 is to enable the learning of a possibly more fair multi-step best respond, i.e. investigate the possibility of learning a correlated equilibrium where the two agents alternate between accessing and yielding. The Q table is updated according to Equation 5:

$$Q(s,a) = \alpha(u + \delta \max_{a'} Q(s',a')) + (1-\alpha)Q(s,a) \tag{5}$$

where α is the learning rate, δ the discount factor, and s, s', a, u the state, next state, action, and utility (reward) respectively. Both approaches select their actions according to an ϵ-greedy policy (as in (Littman and Stone 2002)), i.e. in state s, with probability ϵ they choose a random action, while with probability $1 - \epsilon$ they take action $\arg\max_a Q(s,a)$. The algorithm's performance is highly sensitive to the aforementioned parameters. We have identified two interesting scenarios, presented in Figure 5. Setting $\alpha = 0.75$ and $\epsilon = 0.01$ results in higher efficiency and lower variance, but lower fairness ($\mathbb{J}_{Q_1}(\mathbf{w}) = 0.64$, and $\mathbb{J}_{Q_2}(\mathbf{w}) = 0.83$). On the other hand, $\alpha = 0.5$ and $\epsilon = 0.075$ results in lower efficiency and higher variance (due to the increased randomness), but higher fairness ($\mathbb{J}_{Q_1}(\mathbf{w}) = 0.82$, and $\mathbb{J}_{Q_2}(\mathbf{w}) = 0.89$). The above are true for both approaches (Q_1, and Q_2).

The aforementioned results of Figure 5 suggest that by incorporating a larger state space (i.e. using the common context and the former action) we can achieve better results than

[4] See (Gintis 2000) for an introduction to EGT.

the replicator dynamics indicated. Given a broad enough state space, Q-learning can learn a multi step best response (Littman and Stone 2002). Nevertheless, in both cases, both approaches require a significant number of time-steps to reach a high utilization state. As such, reinforcement learning in anti-coordination scenarios faces similar shortcomings as bandit algorithms, albeit it seems to achieve higher fairness in the evaluated example. Furthermore, it is worth noting that basic reinforcement learning algorithms like Q-learning, compute quantity values for each possible state or state-action pair. As mentioned, mapping anti-coordination to the consensus problem results in an exponential expansion of the solution space, thus in an exponential increase of the computational and memory complexity for the reinforcement learning algorithms as well. The latter constitutes such approaches infeasible for real-world applications.

Instantiations of a correlated equilibrium can be achieved via reinforcement learning. One example is Correlated Q-learning (Greenwald, Hall, and Serrano 2003), albeit it requires the sharing of Q-tables amongst the agents. The latter necessitates either to allow full observability, or a central planner, neither of which is feasible in ah-hoc scenarios.

3.3 Symmetric Strategies & The Price of Anonymity

The two agent resource allocation game of Figure 2 is an inherently symmetric game, yet the only efficient Nash equilibria are asymmetric; one agent yields while the other accesses, achieving 100% efficiency. Asymmetric equilibria of symmetric games are undesirable for two reasons. First, they are unfair and second they require possibly identical agents to differentiate their actions (and thus learning rules). The symmetric MNE (access with probability $\frac{1}{|\zeta|+1}$) on the other hand achieves 0% efficiency. The Price of Anonymity (Cigler and Faltings 2014) allows us to measure the degradation of the system's efficiency (social welfare) due to the requirement of symmetry imposed by anonymity. In an anonymous game agents do not distinguish between other agents, i.e. agents have different utilities but an agent's utility depends only on its own strategy and the number of other agents that chose the same strategy, and not on their identities (Nisan et al. 2007). The Price of Anonymity is the ratio between the optimal social payoff of any (possibly asymmetric) equilibrium and the expected social payoff of the worst symmetric equilibrium. In this example, the price of anonymity is infinite. Nevertheless, it is possible to have solution concepts that are symmetric and efficient by making use of correlated equilibria (Aumann 1974).

Cigler and Faltings developed a symmetric learning rule for reaching an efficient and fair correlated equilibrium of the repeated resource allocation game (Cigler and Faltings 2013). By exploiting the history of their interactions along with the environmental context as a correlation mechanism, the agents are able to learn to coordinate their accesses. Each agent n has a strategy $g_n : \mathcal{K} \rightarrow \{0\} \cup \mathcal{R}$ which maps context to resources. As the algorithm progresses, agents who have successfully accessed a resource $(u_n(a_n, a_{-n}) = 1)$ for a given context value $k \in \mathcal{K}$ will continue to access the same resource every time they observe the same context k. Agents who have not accessed a resource for a given context value k will not attempt to access an occupied resource. If there is a collision, the colliding parties will back-off with probability $p_n^{backoff}$. Algorithm 1 provides the pseudo-code of the allocation algorithm.

Algorithm 1 Pseudo-code of (Cigler and Faltings 2013).

```
Require: $\forall n \in \mathcal{N}$ initialize $g_n$ u.a.r. in $\mathcal{R}$.
Agents observe context $k_t \in \mathcal{K}$.
if $g_n(k_t) > 0$ then
    Agent $n$ accesses resource $r \leftarrow g_n(k_t)$.
    if Collision($r$) then
        Set $g_n(k_t) \leftarrow 0$ with probability $p_n^{backoff}$.
    end if
else if $g_n(k_t) = 0$ then
    Agent $n$ monitors random resource $r \in \mathcal{R}$.
    if Free($r$) then
        Set $g_n(k_t) \leftarrow r$ with probability 1.
    end if
end if
```

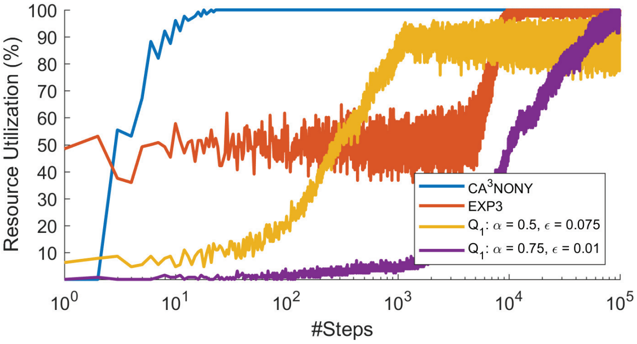


Figure 6: Resource utilization over time of CA^{3}NONY vs. EXP3, Q_1, and Q_2 in the 1-resource, 2-agents allocation game of Figure 2 (x-axis in logarithmic scale).

Algorithm 1 is only applicable in cooperative scenarios. A self-interested agent could stubbornly keep accessing a resource forever, until everyone else backs off (also known as 'bully' strategy (Littman and Stone 2002)[5]). There exist equilibrium back-off probabilities, but in order to actually play them, the agents need to be able to calculate them. It is not always possible to obtain the closed form of the back-off probability distribution of each resource. We have build upon the ideas of (Cigler and Faltings 2013) and proposed instead the adoption of a human-inspired convention of courtesy, which prescribes a constant positive back-off probability in case of collision ($p_n^{backoff} = p > 0, \forall n \in \mathcal{N}$). Coupled with a bookkeeping scheme and punishments for deviating agents, we have proven that adhering to the algorithm is a best-response strategy at each sub-game of the original stage game, given any history of the play. The developed an anti-coordination framework (CA^{3}NONY (Danas-

[5] Such strategies similarly affect Q-learning (Littman and Stone 2002) and bandit algorithms.

sis and Faltings 2018)) still follows to the simple learning rule of Algorithm 1, which allows for fast convergence and its applicability to large scale multi-agent systems.

To verify its performance, Figure 6 depicts the total utilization of resources for the simple 1-resource, 2-agents allocation game of Figure 2, while Figure 7 compares the convergence time of CA^{3}NONY to the fastest of the presented algorithms (EXP3, Q_1, and Q_2) for increasing number of resources R ($N = 2 \times R$). In every case we report the average value over 128 runs of the same simulation. Note that in the first graph, the x-axis is in logarithmic scale, while the second graph is in double logarithmic scale and the error bars represent one standard deviation of uncertainty. For the second simulation (Figure 7), we chose a high enough time horizon ($= 10^8$) to facilitate EXP3 in achieving the convergence criterion ($\geq 90\%$ efficiency) in larger simulations ($R > 64$). Nevertheless, it was unable to do so for $R > 256$, hence the gaps in the EXP3's lines in Figure 7. For the same reason (again regarding Figure 7), we set Q_1 and Q_2's parameters as $\alpha = 0.75$ and $\epsilon = 0.001$. The high learning rate and low randomness were necessary, otherwise Q_1 and Q_2 were unable to reach high utilization. As depicted, CA^{3}NONY is significantly faster than both the bandit and Q-learning algorithms, exhibits lower variance, and can gracefully handle increasing number of resources. In addition to being efficient, CA^{3}NONY converges to a fair allocation $\mathbb{J}_{\text{CA}^3\text{NONY}}(\mathbf{w}) = 1$. Fairness plays an important role, especially in scenarios with scarcity of resources. If the final allocation is fair, rational agents will be more willing to adhere to the protocol and wait for their turn. Under low fairness, the competition between rational agents is increased, which in turn slows down convergence. In Figure 7, the two Q-learning approaches (especially Q_1) might look appealing from the perspective of scalability, but both result in considerably low fairness (lower on average than an unfair PNE). For any number of resources, $\mathbb{J}_{Q_1}(\mathbf{w}) \in [0.45, 0.52]$, with a mean value of 0.48, while $\mathbb{J}_{Q_2}(\mathbf{w}) \in [0.37, 0.48]$, with a mean value of 0.44. Thus, both Q-learning approaches converge to a situation similar to an unfair PNE. In repeated games though, rational agents might not be willing to concede to a PNE (as in the 'bully' strategy of (Littman and Stone 2002)). Finally, CA^{3}NONY provides higher average payoff for the agents (45.09 for CA^{3}NONY vs. -50.54 for the EXP3, -79.03 for Q_1, and -84.08 for Q_2 in the scenario of Figure 6, assuming collision cost $\zeta = -1$), which is an essential indicator of the algorithms individual performance. The latter constitute CA^{3}NONY a promising framework for real-life applications.

4 Conclusion

The relevance of anti-coordination in multi-agent scenarios stems from the need of sharing (possibly) indivisible, limited resources. The curse of dimensionality encompassing the mapping of anti-coordination problems to the classical consensus problem along with the non-stationarity arisen from the simultaneous learning of all the participants make achieving a desirable outcome even more challenging. Furthermore, contrary to coordination problems which are typically encountered in cooperative settings, anti-coordination deals mostly with self-interested, rational agents. Rational agents are able to manipulate the algorithm to maximize their own utility, which brings forth the need for developing algorithms resilient to such manipulations. Ultimately, anti-coordination boils down to incentivizing participants to systematically and consistently adopt less desirable actions, albeit in a way that ensures high efficiency and fairness in the final outcome.

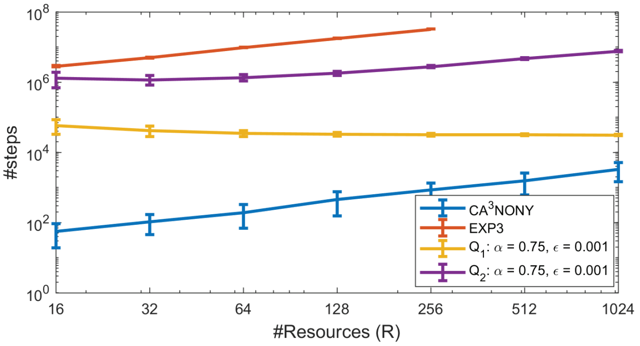


Figure 7: Convergence time of CA^{3}NONY vs. EXP3, Q_1, and Q_2 for increasing number of resources R, $N = 2 \times R$ (double logarithmic scale).

In this paper, we presented a brief overview of multi-agent learning dynamics for the anti-coordination problem, to increase interest and motivate research in the area. We focused on satisfying incentive constraints, efficiency, fairness and convergence speed. Specifically, we examined bandit algorithms, reinforcement learning, and symmetric strategies for the repeated resource allocation game. We demonstrated that most of the classical, well-established multi-agent learning techniques suffer from slow convergence rate and/or poor fairness. An exception to that is CA^{3}NONY , an anti-coordination framework based on the human-inspired convention of courtesy. Contrary to the aforementioned approaches, CA^{3}NONY is able to reach efficient and fair allocations in polynomial time. Moreover, adhering to the protocol constitutes a rational strategy. The latter suggests that human-inspired conventions may prove beneficial in other ad-hoc coordination scenarios as well. An interesting future direction would be to combine well-established multi-agent learning techniques with simple conventions (e.g. allowing others to acquire a resource first (courtesy convention), or maintaining the acquired resource after convergence) for solving more complex anti-coordination problems.

Finally, a generalization of anti-coordination games, called dispersion games, was described in (Grenager, Powers, and Shoham 2002). In a dispersion game, agents are able to choose from several actions, favoring the one that was chosen by the smallest number of agents (analogous to minority games (Challet et al. 2013)). In (Grenager, Powers, and Shoham 2002) the agents do not have any particular preference for the attained equilibrium. Contrary to that, we are interested in achieving an efficient and fair outcome. Expanding the studied techniques to tackle dispersion games, and therefore non-binary utilities, would be another interesting avenue for future research.

References

Albrecht, S. V.; Crandall, J. W.; and Ramamoorthy, S. 2016. Belief and truth in hypothesised behaviours. *Artificial Intelligence* 235:63–94.

Auer, P.; Cesa-Bianchi, N.; Freund, Y.; and Schapire, R. E. 2002. The nonstochastic multiarmed bandit problem. *SIAM journal on computing* 32(1):48–77.

Auer, P.; Cesa-Bianchi, N.; and Fischer, P. 2002. Finite-time analysis of the multiarmed bandit problem. *Machine Learning* 47(2):235–256.

Aumann, R. J. 1974. Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics* 1(1):67 – 96.

Barrett, S., and Stone, P. 2011. Ad hoc teamwork modeled with multi-armed bandits: An extension to discounted infinite rewards. In *Proceedings of 2011 AAMAS Workshop on Adaptive and Learning Agents*, 9–14.

Barrett, S.; Rosenfeld, A.; Kraus, S.; and Stone, P. 2017. Making friends on the fly: Cooperating with new teammates. *Artificial Intelligence* 242:132–171.

Bellman, R. 2013. *Dynamic programming*. Courier Corporation.

Beygelzimer, A.; Langford, J.; Li, L.; Reyzin, L.; and Schapire, R. 2011. Contextual bandit algorithms with supervised learning guarantees. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 19–26.

Bloembergen, D.; Tuyls, K.; Hennes, D.; and Kaisers, M. 2015. Evolutionary dynamics of multi-agent learning: A survey. *J. Artif. Int. Res.* 53(1):659–697.

Busoniu, L.; Babuska, R.; and De Schutter, B. 2008. A comprehensive survey of multiagent reinforcement learning. *Trans. Sys. Man Cyber Part C* 38(2):156–172.

Chakraborty, M.; Chua, K. Y. P.; Das, S.; and Juba, B. 2017. Coordinated versus decentralized exploration in multi-agent multi-armed bandits. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 164–170.

Challet, D.; Marsili, M.; Zhang, Y.-C.; et al. 2013. Minority games: interacting agents in financial markets. *OUP Catalogue*.

Cigler, L., and Faltings, B. 2013. Decentralized anti-coordination through multi-agent learning. *Journal of Artificial Intelligence Research* 47:441–473.

Cigler, L., and Faltings, B. 2014. Symmetric subgame-perfect equilibria in resource allocation. *J. Artif. Int. Res.* 49(1):323–361.

Cigler, L. 2013. *Multi-Agent Learning for Resource Allocation Problems*. Ph.D. Dissertation, École Polytechnique Fédérale de Lausanne.

Coulouris, G. F.; Dollimore, J.; and Kindberg, T. 2005. *Distributed systems: concepts and design*. pearson education.

Danassis, P., and Faltings, B. 2018. A courteous learning rule for ad-hoc anti-coordination. *arXiv:1801.07140*.

Gintis, H. 2000. *Game theory evolving: A problem-centered introduction to modeling strategic behavior*. Princeton university press.

Greenwald, A.; Hall, K.; and Serrano, R. 2003. Correlated q-learning. In *ICML*, volume 3, 242–249.

Grenager, T.; Powers, R.; and Shoham, Y. 2002. Dispersion games: general definitions and some specific learning results. In *AAAI/IAAI*, 398–403.

Jain, R.; Chiu, D.; and Hawe, W. 1998. A quantitative measure of fairness and discrimination for resource allocation in shared computer systems. *CoRR* cs.NI/9809099.

Littman, M. L., and Stone, P. 2002. *Implicit Negotiation in Repeated Games*. Berlin, Heidelberg: Springer Berlin Heidelberg. 393–404.

Nisan, N.; Roughgarden, T.; Tardos, E.; and Vazirani, V. V. 2007. *Algorithmic game theory*, volume 1. Cambridge University Press Cambridge.

Papadimitriou, C. H., and Roughgarden, T. 2008. Computing correlated equilibria in multi-player games. *J. ACM* 55(3):14:1–14:29.

Stone, P.; Kaminka, G. A.; Kraus, S.; and Rosenschein, J. S. 2010. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the Twenty-Fourth Conference on Artificial Intelligence*.

Tuyls, K., and Parsons, S. 2007. What evolutionary game theory tells us about multiagent learning. *Artificial Intelligence* 171(7):406 – 416. Foundations of Multi-Agent Learning.

Watkins, C. J. C. H., and Dayan, P. 1992. Q-learning. *Machine Learning* 8(3):279–292.

Zhou, L. 2015. A survey on contextual multi-armed bandits. *arXiv preprint arXiv:1508.03326*.

The 2018 AAAI Spring Symposium Series

Learning Against Non-Stationary Agents with Opponent Modelling and Deep Reinforcement Learning

Richard Everett, Stephen Roberts
Department of Engineering Science
University of Oxford
{richard, sjrob}@robots.ox.ac.uk

Abstract

Humans, like all animals, both cooperate and compete with each other. Through these interactions we learn to observe, act, and manipulate to maximise our utility function, and continue doing so as others learn with us. This is a decentralised non-stationary learning problem, where to survive and flourish an agent must adapt to the gradual changes of other agents as they learn, as well as capitalise on sudden shifts in their behaviour. To learn in the presence of such non-stationarity, we introduce the *Switching Agent Model* (SAM) that combines traditional deep reinforcement learning – which typically performs poorly in such settings – with opponent modelling, using uncertainty estimations to robustly switch between multiple policies. We empirically show the success of our approach in a multi-agent continuous-action environment, demonstrating SAM's ability to identify, track, and adapt to gradual and sudden changes in the behaviour of non-stationary agents.

1 Introduction

Cooperation and competition are the cornerstones of both human and animal societies, appearing deeply embedded in our understanding of social intelligence. For an individual agent to maximise their utility function in these societies, they must learn to interact with and against others, as well as understand the consequences of their actions. Studies on this interaction have a long history across domains including game theory (Rapoport and Chammah 1965), evolutionary biology (Strassmann et al. 2011), and multi-agent systems (Shoham, Powers, and Grenager 2007).

A key component of learning to interact with others is the ability to reason about their behaviour. This is accomplished through constructing and utilising models of their decision-making processes, and is commonly referred to as *opponent modelling*. Due to recent advances in machine learning, autonomous agents are increasingly interacting with others, be it negotiating with humans (Lewis et al. 2017) or communicating with other agents (Foerster et al. 2016). It is therefore important that they are able to take into account more than just their own agency.

If done correctly, these models can be used to derive an optimal policy against an agent, such as by exploiting their suboptimal behaviour to yield a higher reward (Ganzfried and Sandholm 2011). However, the construction of these models is non-trivial as an agent's behaviour is rarely stationary, thus requiring any learned opponent model to be continuously updated (Hernandez-Leal et al. 2017b). For example, such non-stationarity can occur in human-computer interaction, whereby a user's behaviour changes gradually as they become familiar with a system, as well as suddenly when the user switches with another user. Likewise, in a competitive setting an agent may initially learn before switching to an alternative strategy as its beliefs about the world change.

One of the most successful paradigms for learning a policy is reinforcement learning (Sutton and Barto 1998), whereby agents learn to maximise their cumulative long-term reward through trial-and-error interactions with their environment. In recent years, the area has experienced a string of successes in the single-agent domain due to advances in deep learning (Mnih et al. 2015). However, the non-stationarity that arises from interacting with other agents renders many single-agent algorithms unsuitable (Hernandez-Leal et al. 2017a). By not taking into account the agency of others, traditional deep reinforcement learning methods struggle to transfer their successes to the multi-agent setting.

Within deep reinforcement learning, opponent modelling has started to receive increasing attention. It has been successfully applied to modelling the policy of agents which switch between episodes (He et al. 2016), albeit requiring handcrafted behavioural features, and more recently it has been used to learn approximate policies of learning agents (Foerster et al. 2017; Lowe et al. 2017). However, to date there has been little progress made on folding in uncertainty into these deep models – a vital component for a robust opponent model – and both forms of non-stationarity have been rarely considered.

In this work, we consider the setting where multiple independent non-stationary agents interact in the same environment. Specifically, we look at two distinct forms of non-stationary behaviour:

1. *Sudden* changes, whereby an agent switches between a set of behaviours through time.
2. *Gradual* changes, whereby an agent's behaviour slowly adjusts over time as it learns.

To learn and succeed in this setting, we propose the *Switching Agent Model* (SAM). By combining traditional deep reinforcement learning algorithms with models of the decision-making processes of other agents, our method learns a general policy which is more robust and adaptive to non-stationary agents. We achieve this combination by explicitly learning from an agent's state-action trajectory with an approximate Bayesian neural network, using Monte Carlo dropout (Gal and Ghahramani 2016) to obtain predictive uncertainty of our model's predictions. The model's predictive error and uncertainty are tracked by a switchboard to robustly identify changes in an opposing agent's behaviour, switching between opponent models and their associated policies through time.

We demonstrate the capabilities of SAM through two experiments in a multi-agent continuous-action environment. First, we show that our approach can identify, track, and adapt to the behaviour of an agent which switches between policies over time, outperforming traditional deep reinforcement learning. Next, we show that the same method also helps in the presence of a learning agent, yielding a higher performance as a result. We finish with an analysis into the uncertainty of our learned opponent models throughout training in both experiments.

2 Switching Agent Model (SAM)

The motivation behind our work is that opposing agents can change their behaviour through time, and therefore our ability to derive an optimal policy in their presence depends on how well we can identify and track this change.

To track these changes and learn an appropriate response, we propose the *Switching Agent Model*. SAM is a collection of inferred opposing agent policies $\hat{\mu}$ and associated approximate best-response policies μ which are connected through, and controlled by, a switchboard. In the following sections we describe each of these components in detail. To aid readability, we refer to inferred opposing agent policies as 'opponent models' and learned approximate best-response policies as 'response policies'. Furthermore, to ease notation we omit the parameters θ and ϕ of μ and $\hat{\mu}$ respectively.

2.1 Switchboard

At the heart of SAM is the switchboard. It tracks the performance of opponent models through time, switching between them and their associated response policies as the opposing agent adjusts its behaviour.

As we consider agents which can adapt both gradually and suddenly, it is important that our switching mechanism can operate in the presence of both changes. To achieve this, our switchboard tracks the running error of the opponent models, expecting notable spikes when an agent switches behaviour and a gradual accumulation over time as they learn. In both of these situations, the value of this running error can be used to initiate a switch. We describe this switching process here and also present it in Algorithm 1.

While an opponent model $\hat{\mu}^k$ is active, the switchboard monitors its running error r. At each timestep t, the active model predicts the next action of the opposing agent using Monte Carlo dropout, obtaining an action prediction $\hat{a}_t^j$ along with its associated predictive uncertainty η_t. On the following timestep $t+1$, the true action a_t^j is observed and the running error is updated as follows:

$$r = r + \frac{|a_t^j - \hat{a}_t^j|}{\eta_t}. \tag{1}$$

If the running error r is less than the specified switch threshold, i.e. $r < r_{\max}$, then the running error is decayed by d. Otherwise, a switch occurs whereby a different opponent model is chosen and the running error is reset, $r \leftarrow 0$.

Algorithm 1: Model Switching Algorithm

```
Input: opponent models $\hat{\mu} = \{\hat{\mu}^1, ..., \hat{\mu}^K\}$, error
       threshold $r_{\max}$, error decay $d$, predict action
       parameters $Z = (N, p, \mathcal{N})$
Initialise running error $r \leftarrow 0$
Initialise current opponent model index $k \leftarrow 1$
for episode $= 1$ to ... do
    Receive initial state $s_0$
    $\hat{a}_0, \eta_0 \leftarrow$ Predict($s_0, \hat{\mu}^k, Z$) ; // Algorithm 2
    for timestep $t = 1$ to ... do
        Receive state $s_t$ and action $a_{t-1}$
        Update running error $r$ using $(a_{t-1}, \hat{a}_{t-1}, \eta_{t-1})$
        if $r >= r_{\max}$ then
            Switch to different opponent model, updating
              $k$
            Reset rolling error $r \leftarrow 0$
        else
            Decay rolling error $r \leftarrow r - d$
        end
        Train $\hat{\mu}^k$ on $(s_{t-1}, a_{t-1})$
        $\hat{a}_t, \eta_t \leftarrow$ Predict($s_t, \hat{\mu}^k, Z$)
    end
end
```

Similar to related approaches which consider switching behaviours (Hernandez-Leal et al. 2016), we assume that the modelled agent will not switch while our method is initially learning an opponent model.

2.2 Response Policies

To learn a policy which can act in an environment, as well as take advantage of our inferred opponent models, we use the Deep Deterministic Policy Gradient (DDPG) algorithm (Lillicrap et al. 2015). We refer to these as 'response' policies due to their association with a specific opponent model.

By alternating between models over time according to the opposing agent's behaviour, and therefore alternating between response policies, our general policy is comprised of multiple sub-policies $\mu = \{\mu^1, ..., \mu^K\}$.

Each policy μ^k is trained by sampling from its own replay buffer $\mathcal{D}^k$, learning an approximate best-response to the historical average behaviour of the opponent contained within the buffer. The learned response policies are then used by the agent to select actions given their observed state s_t^i and the predicted next actions from the associated opponent model $\hat{a}_t = \hat{\mu}^k(s_t^i)$, plus some optional noise from an exploration

Algorithm 2: Predict Action Algorithm

```
Input: state $s$, opponent policy $\hat{\mu}$, Z=(number of
       passes $N$, dropout probability $p$, inherent
       noise $\mathcal{N}$)
Output: action prediction $\hat{a}$, predictive uncertainty $\eta$
for $n = 1$ to $N$ do
    $\hat{a}_n \leftarrow \hat{\mu}(s)$ ;            // with dropout (p)
end
$\hat{a} \leftarrow \frac{1}{N}\sum_{n=1}^{N} \hat{a}_n$ ;          // model prediction
$\hat{\sigma}^2 \leftarrow \frac{1}{N}\sum_{n=1}^{N}(\hat{a}_n - \hat{a})^2$ ; // model uncertainty
$\eta \leftarrow \sqrt{\hat{\sigma}^2 + \text{Var}(\mathcal{N})}$ ;              // predictive
 uncertainty
```

process:

$$a_t = \mu^k(s_t^i \| \hat{a}_t) + \mathcal{N}_t. \tag{2}$$

2.3 Opponent Models

The final component of SAM is the set of inferred opposing agent policies, i.e. opponent models $\hat{\mu}$, which are learned from observed state-action trajectories and switched between across time by the switchboard.

Our method takes advantage of these learned opponent models in two distinct ways. First, as shown in Equation 2, we input their predictions of the agent's next actions directly into our response policy. Depending on the quality of our models, this can help reduce the perceived non-stationarity of the environment. Second, we use the model's uncertainty estimations in its predictions to obtain a measure of normal and unexpected behaviour, allowing our switchboard to change models accordingly.

As our agent acts in a domain with continuous actions, predicting an agent's next action is a regression problem. In this context, our model's uncertainty can be thought of as a confidence interval around its predictions.

To learn an approximation of agent j's true policy μ_j^k, we use a neural network $\hat{\mu}_j^k$ parametrised by ϕ_j^k which we optimise by minimising the loss:

$$L(\phi_j^k) = \left(\hat{\mu}^k(s_{t-1}^i) - a_{t-1}^j\right)^2, \tag{3}$$

where s_{t-1}^i is the previously observed state by agent i and a_{t-1}^j is the true action which agent j performed.

To identify changes in the behaviour of an agent, our opponent models need a measure of uncertainty to detect and assign error to unexpected actions. We do this by approximating our model's predictive uncertainty using Monte Carlo dropout (Gal and Ghahramani 2016). The process for this is presented in Algorithm 2 and described below.

First, to obtain our model's prediction of agent j's true next action a_{t+1}^j we pass the observing agent's state s_t^i through the network N times, where on each pass the output is computed by randomly dropping out each hidden unit with probability p:

$$\hat{a}_{t+1}^j = \frac{1}{N}\sum_{n=1}^{N} \hat{\mu}^k(s_t^i). \tag{4}$$

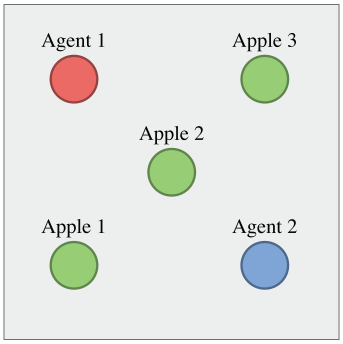


Figure 1: Gathering environment.

Next, to obtain predictive uncertainty we first determine our model's uncertainty which can be approximated using the sample variance:

$$\hat{\sigma^2}(\hat{\mu}^k(s_t^i)) = \frac{1}{N}\sum_{n=1}^{N}\left(\hat{\mu}^k(s_t^i) - \hat{a}_{t+1}^j\right)^2. \tag{5}$$

Our complete predictive uncertainty is thus given by:

$$\eta = \sqrt{\hat{\sigma^2}(\hat{\mu}^k(s_t^i)) + \text{Var}(\mathcal{N})}, \tag{6}$$

where $\mathcal{N}$ is the inherent noise in the agent's actions.

Both of the outputs of this process, namely the prediction $\hat{a}$ and its predictive uncertainty η, are used by the switchboard to determine the running error r of the current opponent model (see Equation 1).

3 Experiments

In our experiments, we train an agent against a non-stationary adversary with the aim of showing that traditional reinforcement learning performs sub-optimally in such a setting, and that our proposed methods – i.e. deep opponent modelling with uncertainty estimates and multiple policies – help improve performance.

We compare the performance of SAM and DDPG. This comparison is done indirectly in our first experiment, placing both agents against a separate switching adversary, while in second experiment we compare them directly.

3.1 Experimental Details

Environment: Both of our experiments take place in a gathering scenario between two independent agents in a continuous world populated by apples as shown in Figure 1.

In this scenario, agents can receive a reward in two ways:

1. Collecting apples by moving onto them, yielding a small reward.
2. Stealing from another agent by colliding into them, yielding a larger reward but penalising the other agent.

Under this reward structure, an agent's behaviour can be characterised along a scale ranging from purely cooperative (i.e. never stealing) to purely defective (i.e. only stealing). An agent's optimal behaviour along this scale depends on the behaviour of their opposing agent. For example, against a purely cooperative agent it is desirable to steal more, while against a purely defective agent it is advantageous to focus on avoiding them while collecting apples as fast as possible.

At each timestep, agents observe their own position and velocity, their relative position and velocity to the opposing agent, their relative position to each apple, and whether they can steal. In addition, they can move around the environment by exerting a continuous force in two directions.

To help speed up training, each agent is given a small negative reward proportional to their distance to the nearest apple, as well as a small negative reward proportional to their distance to the opposing agent when it is possible to steal.

Model Architecture: The actor and critic networks in our response policies have 2 hidden layers with 64 hidden units in each layer, ReLU activation functions, and tanh on the actor's output layer. Opponent model networks consist of 3 hidden layers with 64 hidden units in each layer, ReLU activation functions, and tanh on the output layer.

Algorithm Parameters: We train using Adam (Kingma and Ba 2014) with a learning rate of 0.0001 and 0.001 for the actors and critics respectively. Soft target updates are done with $\tau = 0.01$. We set the discount factor to $\gamma = 0.95$, store 10^6 transitions in our replay buffer (which are equally distributed between the K buffers in SAM), and train on mini-batches of 64 transitions. During training, exploration is achieved by adding noise using an Ornstein-Uhlenbeck process with $\sigma = 0.1$.

Our opponent models are trained with a learning rate of 0.001. Predictive uncertainty is obtained using $N = 30$ forward passes with dropout probability $p = 0.10$, and errors are decayed by $d = 5$ with the switch threshold set to $r_{\max} = 50$. Additionally, we set $K = 2$ (i.e. two opponent models and two response policies), and therefore switch calls alternate k between 1 and 2.

Train & Test Procedure: For training, we segment timesteps into episodes of 1,000 timesteps, resetting the environment at the start of each episode. Agents sample a minibatch of transitions to learn from at each timestep, and actions are selected with an added exploration noise. Between training episodes we test agents for a further 1,000 steps, selecting actions with no exploration noise.

3.2 Experiment 1: Sudden Changes

In this experiment, we consider the setting of a learning agent competing against a non-stationary adversary that switches between policies throughout time. To perform optimally in this setting, an agent needs to be able to quickly identify when the adversary has switched behaviours and respond accordingly.

Switching Adversary To construct the policies used by the switching adversary, we train two agents with different reward schedules:

1. Passive: rewarded for collecting apples and penalised for collisions.

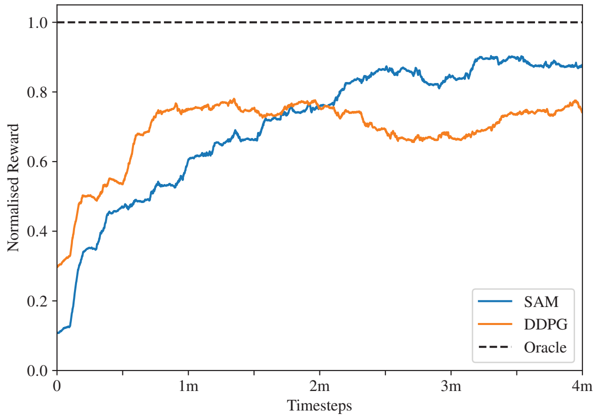


Figure 2: Average reward of SAM (blue) and DDPG (orange) against the switching adversary. Normalised by the oracle's best performance (black dashed line) and smoothed across switches to improve readability. Average of 5 trials.

2. Offensive: only rewarded for stealing apples from the opposing agent.

The switching adversary is made non-stationary by switching between these two learned policies, whereby the agent follows one policy for a number of timesteps before switching to the other, repeating this process through time. To meet our assumption mentioned in Section 2.1, we enforce a minimum number of timesteps between switches.

As the switching adversary's behaviour changes through time, so too does the optimal response. Specifically, the optimal strategy against a passive agent is to steal apples, while the optimal strategy against an offensive agent is to avoid them while collecting apples.

When comparing agents against the switching adversary, we normalise their performance metrics by the highest metrics achieved by an agent with access to true state of the switching adversary (known as the oracle).

Results In Figure 2, we present the reward of each agent throughout training, normalising by the performance of the oracle. As can be seen, there are clear differences between the performances of the two agents.

Due to the DDPG agent learning one response policy rather than two, it can reuse what is has learned in new situations, even if that behaviour is not necessarily optimal. This leads to DDPG quickly performing better than SAM. However, again due to its single policy, it learns a best response to the switching adversary's average behaviour, leading to a suboptimal performance as training continues. In contrast, SAM is able to learn a specific best response to each of the agent's behaviours, yielding an eventual higher reward.

Switch Analysis As our model relies on detecting changes in the opposing agent's behaviour, it is important to know how long it takes to detect a change once is has occurred. To do this, we run our trained opponent model over known change points, which the model does not know, measuring the error rate and logging when a change occurs.

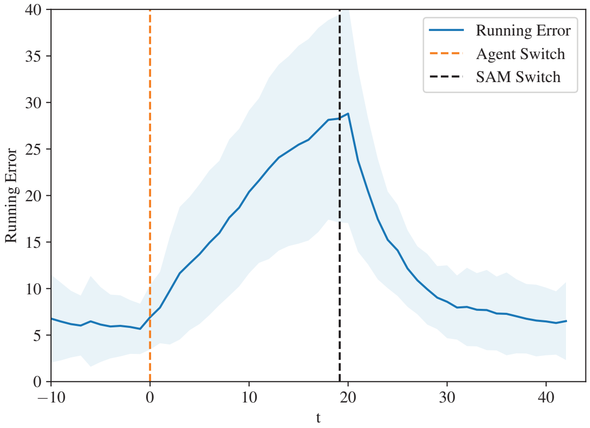


Figure 3: Average running error of our opponent model (blue). Adversary agent switches behaviour at $t = 0$ (dotted orange) and SAM switches around $t = 19$ (dotted black). Results are averaged across 100 examples.

The results of this are presented in Figure 3, whereby we visualise the running error of a learned opponent model for 100 switches, centering the agent's switch on timestep $t = 0$. On average, our method is able to detect a switch in the adversary's behaviour after 19.2 ± 4.4 timesteps as indicated by the dotted black line. In other words, on average we can detect a change in an agent's behaviour from observing ≈19 of their actions.

3.3 Experiment 2: Gradual Changes

In this experiment, we consider the setting of two agents simultaneously learning in the same environment. We train a SAM agent against a DDPG agent, evaluating them after training for 10,000 further timesteps with no exploration noise in their action selection.

Normalising performance metrics by the total achieved by both agents, we find that SAM manages to obtain a higher reward than DDPG (0.57 to 0.43), doing so by both stealing more (0.53 to 0.47) as well as collecting more apples (0.56 to 0.44). Along with our results from Experiment 1, this suggests that our methods help improve learning against a non-stationary agent which not only changes suddenly but also gradually as they learn.

3.4 Uncertainty Analysis

As we use our uncertainty estimations to determine when to switch policies, it is important that they are robust and have meaning across training. In Figure 4, we present our findings on this from the start, middle, and end of training from the previous two experiments.

At the top we visualise our model's predictive uncertainty in the switching adversary's actions along two dimensions, where each ring represents one standard deviation from the zero-centered mean. As can be seen, predictive uncertainty is high at the start of training and markedly low at the end. This is encouraging as it means that our model will return errors with high confidence when the switching adversary starts behaving anomalously compared to our current beliefs, causing a switch to occur. We saw this in practice in our first experiment (see Figure 3).

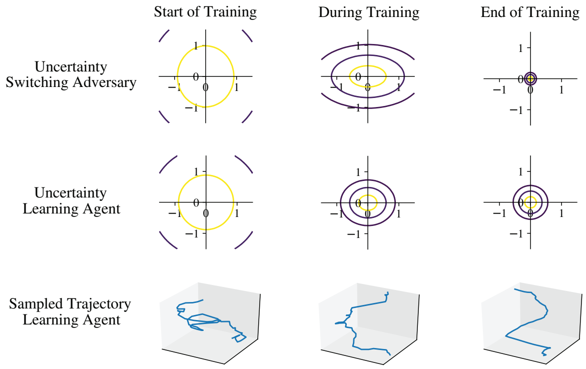


Figure 4: Predictive uncertainty of the opponent model trained on the switching adversary (top) and learning agent (middle) for the same state, along with the learning agent's trajectory (bottom). Columns represent the stage of training from start (left) to end (right).

While it is intuitive that introducing uncertainty will help against a switching adversary, it is less intuitive when applied to an agent who is also learning. To investigate this, we repeat the previous analysis but instead consider a learning agent (middle row) and visualise their trajectory through time (bottom row).

From these rows, we can see that our model's predictive uncertainty decreases as the agent's actions become more consistent and meaningful. In comparison to the switching adversary's result, the predictive uncertainty is higher due to the learning agent selecting actions with some exploration noise. We conclude from this that an agent's trajectory (and therefore their actions) is indicative of the stage of training, and that our opponent models are able to capture this.

4 Conclusions & Future Work

In this work, we proposed the *Switching Agent Model* (SAM) as a way of learning in the presence of non-stationary agent behaviour. We achieved this through combining traditional deep reinforcement learning with opponent modelling, using uncertainty estimations from Monte Carlo dropout to robustly switch between opponent models and their associated response policies. We empirically demonstrated the benefits of our approach in a continuous-action environment against two types of agents and presented insights into the uses of uncertainty.

Future work will further investigate the applicability of opponent modelling in the presence of another deep learning agent. Additionally, we will also investigate the dynamics of our switching strategy in the presence of other types of non-stationary agents, such as those who are actively trying to exploit the switching mechanism, looking at how it compares to alternative strategies.

References

Foerster, J.; Assael, Y. M.; de Freitas, N.; and Whiteson, S. 2016. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2137–2145.

Foerster, J. N.; Chen, R. Y.; Al-Shedivat, M.; Whiteson, S.; Abbeel, P.; and Mordatch, I. 2017. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*.

Gal, Y., and Ghahramani, Z. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, 1050–1059.

Ganzfried, S., and Sandholm, T. 2011. Game theory-based opponent modeling in large imperfect-information games. In *The 10th International Conference on Autonomous Agents and Multiagent Systems-Volume 2*, 533–540. International Foundation for Autonomous Agents and Multiagent Systems.

He, H.; Boyd-Graber, J.; Kwok, K.; and Daumé III, H. 2016. Opponent modeling in deep reinforcement learning. In *International Conference on Machine Learning*, 1804–1813.

Hernandez-Leal, P.; Taylor, M. E.; Rosman, B.; Sucar, L. E.; and Munoz de Cote, E. 2016. Identifying and tracking switching, non-stationary opponents: a bayesian approach.

Hernandez-Leal, P.; Kaisers, M.; Baarslag, T.; and de Cote, E. M. 2017a. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183*.

Hernandez-Leal, P.; Zhan, Y.; Taylor, M. E.; Sucar, L. E.; and de Cote, E. M. 2017b. Efficiently detecting switches against non-stationary opponents. *Autonomous Agents and Multi-Agent Systems* 31(4):767–789.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lewis, M.; Yarats, D.; Dauphin, Y. N.; Parikh, D.; and Batra, D. 2017. Deal or no deal? end-to-end learning for negotiation dialogues. *arXiv preprint arXiv:1706.05125*.

Lillicrap, T. P.; Hunt, J. J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; and Wierstra, D. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Rapoport, A., and Chammah, A. M. 1965. *Prisoner's dilemma: A study in conflict and cooperation*, volume 165. University of Michigan press.

Shoham, Y.; Powers, R.; and Grenager, T. 2007. If multi-agent learning is the answer, what is the question? *Artificial Intelligence* 171(7):365–377.

Strassmann, J. E.; Queller, D. C.; Avise, J. C.; and Ayala, F. J. 2011. In the light of evolution v: Cooperation and conflict.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Bayesian Opponent Exploitation in Imperfect-Information Games

Sam Ganzfried
School of Computing and Information Sciences
Florida International University
sganzfri@cis.fiu.edu

Qingyun Sun
School of Mathematics
Stanford University
qysun@stanford.edu

Abstract

Two fundamental problems in computational game theory are computing a Nash equilibrium and learning to exploit opponents given observations of their play (opponent exploitation). The latter is perhaps even more important than the former: Nash equilibrium does not have a compelling theoretical justification in game classes other than two-player zero-sum, and for all games one can potentially do better by exploiting perceived weaknesses of the opponent than by following a static equilibrium strategy throughout the match. The natural setting for opponent exploitation is the Bayesian setting where we have a prior model that is integrated with observations to create a posterior opponent model that we respond to. The most natural, and a well-studied prior distribution is the Dirichlet distribution. An exact polynomial-time algorithm is known for best-responding to the posterior distribution for an opponent assuming a Dirichlet prior with multinomial sampling in normal-form games; however, for imperfect-information games the best known algorithm is based on approximating an infinite integral without theoretical guarantees. We present the first exact algorithm for a natural class of imperfect-information games. We demonstrate that our algorithm runs quickly in practice and outperforms the best prior approaches. We also present an algorithm for the uniform prior setting.

1 Introduction

Imagine you are playing a game repeatedly against one or more opponents. What algorithm should you use to maximize your performance? The classic "solution concept" in game theory is the Nash equilibrium. In a Nash equilibrium σ, each player is simultaneously maximizing his payoff assuming the opponents all follow their components of σ. So should we just find a Nash equilibrium strategy for ourselves and play it in all the game iterations?

Unfortunately, there are some complications. First, there can exist many Nash equilibria, and if the opponents are not following the same one that we have found (or are not following one at all), then our strategy would have no performance guarantees. Second, finding a Nash equilibrium is challenging computationally: it is PPAD-hard and is widely conjectured that no polynomial-time algorithms exist (Chen and Deng 2006). These challenges apply to both extensive-form games (of both perfect and imperfect information) and strategic-form games, for games with more than two players and non-zero-sum games. While a particular Nash equilibrium may happen to perform well in practice, there is no theoretically compelling justification for why computing one and playing it repeatedly is a good approach. Two-player zero-sum games do not face these challenges: there exist polynomial-time algorithms for computing an equilibrium (Koller, Megiddo, and von Stengel 1994), and there exists a game value that is guaranteed in expectation in the worst case by all equilibrium strategies regardless of the strategy played by the opponent (and this value is the best worst-case guaranteed payoff for any of our strategies). However, even for this game class it would be desirable to deviate from equilibrium in order to learn and exploit perceived weaknesses of the opponent; for instance, if the opponent has played Rock in each of the first thousand iterations of rock-paper-scissors, it seems desirable to put additional weight on paper beyond the equilibrium value of $\frac{1}{3}$.

Thus, learning to exploit opponents' weaknesses is desirable in all game classes. One approach would be to construct an opponent model consisting of a single mixed strategy that we believe the opponent is playing given our observations of his play and a prior distribution (perhaps computed from a database of historical play). This approach has been successfully applied to exploit weak agents in limit Texas hold 'em poker, a large imperfect-information game (Ganzfried and Sandholm 2011). A drawback is that it is potentially not robust. It is very unlikely that the opponent's strategy matches this point estimate exactly, and we could perform poorly if our model is incorrect. A more robust approach, which is the natural one to use in this setting, is to use a Bayesian model, where the prior and posterior are full distributions over mixed strategies of the opponent, not single mixed strategies. A natural prior distribution, which has been studied and applied in this context, is the Dirichlet distribution. The pdf of the Dirichlet distribution is the belief that the probabilities of K rival events are x_i given that each event has been observed $\alpha_i - 1$ times: $f(x, \alpha) = \frac{1}{B(\alpha)} \prod x_i^{\alpha_i - 1}$.[1] Some

[1] $B(\alpha)$ is the beta function $B(\alpha) = \frac{\prod \Gamma(\alpha_i)}{\Gamma\left(\sum_i \alpha_i\right)}$, where $\Gamma(n) = (n-1)!$ is the gamma function.

notable properties are that the mean is $E[X_i] = \frac{\alpha_i}{\sum_k \alpha_k}$ and that, assuming multinomial sampling, the posterior after including new observations is also Dirichlet, with parameters updated based on the new observations.

Prior work has presented an efficient algorithm for optimally exploiting an opponent in normal-form games in the Bayesian setting with a Dirichlet prior (Fudenberg and Levine 1998), which is essentially the fictitious play rule (Brown 1951). Given prior counts α_i for each opponent action, the algorithm increments the counter for an action by one each time it is observed, and then best responds to a model for the opponent where he plays each strategy in proportion to the counters. This algorithm would also extend directly to sequential games of perfect information, where we maintain independent counters at each opponent decision node; this would also work for games of imperfect information where the opponent's private information is observed after each round (so that we would know exactly what information set he took the observed action from). For all of these game classes the algorithm would apply to both zero and general-sum games, for any number of players. However, it would not apply to imperfect-information games where opponents' private information is not observed after play.

An algorithm exists for approximating a Bayesian best response in imperfect-information games, which uses importance sampling to approximate an infinite integral. This algorithm has been successfully applied to limit Texas hold 'em poker (Southey et al. 2005). However, it is only a heuristic approach with no guarantees. The authors state,

> "Computing the integral over opponent strategies depends on the form of the prior but is difficult in any event. For Dirichlet priors, it is possible to compute the posterior exactly but the calculation is expensive except for small games with relatively few observations. This makes the exact BBR an ideal goal rather than a practical approach. For real play, we must consider approximations to BBR."

However, we see no justification for the claim that it is possible to compute the posterior exactly in prior work, and there could easily be no closed-form solution. In this paper we present a solution for this problem, leading to the first exact optimal algorithm for performing Bayesian opponent exploitation in imperfect-information games. While the claim is correct that the computation is expensive for large games, we show that in a small (yet realistic) game it outperforms all prior approaches. Furthermore, we show that the computation can run extremely quickly even for large number of observations (though it can run into numerical instability), contradicting the second claim. We also present general theory, and an algorithm for another natural prior distribution (uniform distribution over a polyhedron).

2 Meta-algorithm

The problem of developing efficient algorithms for optimizing against a posterior distribution, which is a probability distribution over mixed strategies for the opponent (which are themselves distributions over pure strategies) seems daunting. We need to be able to compactly represent the posterior distribution and efficiently compute a best response to it. Fortunately, we show that our payoff of playing any strategy σ_i against a probability distribution over mixed strategies for the opponent equals our payoff of playing σ_i against the mean of the distribution. Thus, we need only represent and respond to the single strategy that is the mean of the distribution, and not to the full distribution. While this result was likely known previously, we have not seen it stated explicitly, and it is important enough to be highlighted so that it is on the radar of the AI community.

Suppose the opponent is playing mixed strategy σ_{-i} where $\sigma_{-i}(s_{-j})$ is the probability that he plays pure strategy $s_{-j} \in S_{-j}$. By definition of expected utility, $u_i(\sigma_i, \sigma_{-i}) = \sum_{s_{-j} \in S_{-j}} \sigma_{-i}(s_{-j}) u_i(\sigma_i, s_{-j})$. We can generalize this naturally to the case where the opponent is playing according to a probability distribution with pdf f_{-i} over mixed strategies: $u_i(\sigma_i, f_{-i}) = \int_{\sigma_{-i} \in \Sigma_{-i}} [f_{-i}(\sigma_{-i}) \cdot u_i(\sigma_i, \sigma_{-i})]$. Let $\overline{f_{-i}}$ denote the mean of f_{-i}. That is, $\overline{f_{-i}}$ is the mixed strategy that selects s_{-j} with probability $\int_{\sigma_{-i} \in \Sigma_{-i}} [\sigma_{-i}(s_{-j}) \cdot f_{-i}(\sigma_{-i})]$. Then we have the following:

Theorem 1.
$$u_i(\sigma_i, \overline{f_{-i}}) = u_i(\sigma_i, f_{-i}).$$

That is, the payoff against the mean of a strategy distribution equals the payoff against the full distribution.

Proof.

$$\begin{aligned}
& u_i(\sigma_i, \overline{f_{-i}}) \\
= \; & \sum_{s_{-j} \in S_{-j}} \left[u_i(\sigma_i, s_{-j}) \int_{\sigma_{-i} \in \Sigma_{-i}} [\sigma_{-i}(s_{-j}) \cdot f_{-i}(\sigma_{-i})] \right] \\
= \; & \sum_{s_{-j} \in S_{-j}} \left[\int_{\sigma_{-i} \in \Sigma_{-i}} [u_i(\sigma_i, s_{-j}) \cdot \sigma_{-i}(s_{-j}) \cdot f_{-i}(\sigma_{-i})] \right] \\
= \; & \int_{\sigma_{-i} \in \Sigma_{-i}} \left[\sum_{j \in S_{-j}} [u_i(\sigma_i, s_{-j}) \cdot \sigma_{-i}(s_{-j}) \cdot f_{-i}(\sigma_{-i})] \right] \\
= \; & \int_{\sigma_{-i} \in \Sigma_{-i}} [u_i(\sigma_i, \sigma_{-i}) \cdot f_{-i}(\sigma_{-i})] \\
= \; & u_i(\sigma_i, f_{-i})
\end{aligned}$$

□

Theorem 1 applies to both normal and extensive-form games (with perfect or imperfect information), for any number of players (σ_{-i} could be a joint strategy profile for all opposing agents).

Now suppose the opponent is playing according a prior distribution $p(\sigma_{-i})$, and let $p(\sigma_{-i}|x)$ denote the posterior probability given observations x. Let $\overline{p(\sigma_{-i}|x)}$ denote the mean of $p(\sigma_{-i}|x)$. As an immediate consequence of Theorem 1, we have the following corollary.

Corollary 1. $u_i(\sigma_i, \overline{p(\sigma_{-i}|x)}) = u_i(\sigma_i, p(\sigma_{-i}|x))$.

Corollary 1 implies the meta-procedure for optimizing performance against an opponent using p:

There are several challenges for applying Algorithm 1. First, it assumes that we can compactly represent the prior and posterior distributions p_t, which have infinite domain

Algorithm 1 Meta-algorithm for Bayesian opponent exploitation

```
Inputs: Prior distribution $p_0$, response functions $r_t$ for $0 \leq t \leq T$
    $M_0 \leftarrow \overline{p_0(\sigma_{-i})}$
    $R_0 \leftarrow r_0(M_0)$
    Play according to $R_0$
    for $t = 1$ to $T$ do
        $x_t \leftarrow$ observations of opponent's play at time step $t$
        $p_t \leftarrow$ posterior distribution of opponent's strategy given prior $p_{t-1}$ and observations $x_t$
        $M_t \leftarrow$ mean of $p_t$
        $R_t \leftarrow r_t(M_t)$
        Play according to $R_t$
```

(the set of opponents' mixed strategy profiles). Second, it requires a procedure to efficiently compute the posterior distributions given the prior and the observations, which requires updating potentially infinitely many strategies. Third, it requires an efficient procedure to compute the mean of p_t. And fourth, it requires that the full posterior distribution from one round be compactly represented to be used as the prior in the next round. We can address the fourth challenge by using a modified update step:

$p_t \quad \leftarrow \quad$ posterior distribution of opponent's strategy given prior p_0 and observations $x_1, \ldots, x_t$.

We will be using this new rule in our main algorithm.

The response functions r_t (which return a strategy for ourselves that performs well against input opponents' strategies) could be standard best response, for which linear-time algorithms exist in games of imperfect information (and a recent approach has enabled efficient computation in extremely large games (Johanson et al. 2011)). They could also be a more robust response, e.g., one that places a limit on the exploitability of our own strategy, perhaps one that varies over time based on performance (or a lower-variance estimator) (Johanson, Zinkevich, and Bowling 2007; Johanson and Bowling 2009; Ganzfried and Sandholm 2015). In particular, the restricted Nash response has been demonstrated to outperform best response against agents in limit Texas hold 'em whose actual strategy may differ substantially from the exact model (Johanson, Zinkevich, and Bowling 2007).

3 Robustness of the approach

It has been pointed out that, empirically, the approach described is not robust: if we play a full best response to a point estimate of the opponent's strategy we can have very high exploitability ourselves, and could perform very poorly if in fact we are wrong about our model (Johanson, Zinkevich, and Bowling 2007). This could happen for several reasons. Our modeling algorithm could be incorrect: it could make an incorrect assumption about the prior and form of the opponent's distribution. This could happen because the opponent changes his strategy over time (possibly either by improving his own play or by adapting to our play), in which case a model that assumes a static opponent could be predicting a strategy that the opponent is no longer using. The opponent could also have modified his play strategically in an attempt to deceive us (e.g., the opponent initially starts off playing extremely conservatively, then switches to a more aggressive style as he suspects we try to exploit his conservatism).

A second reason that we could be wrong in our opponent model other than our modeling algorithm incorrectly modeling the opponents' dynamic approach is that our observations of his play are very noisy (due to both randomization in the opponent's strategy and to the private information selected by chance), particularly over a small sample. Even if our approach is correct and the opponent is in fact playing a static strategy according to the distribution assumed by the modeling algorithm, it is very unlikely that our actual perception of his strategy is precisely correct. A third reason, of course, is that the opponent may be following a static strategy that does not exactly conform to our model for the prior and/or sampling method used to generate the posterior.

Suppose we believe the opponent is playing x_{-i}, while he is actually playing x'_{-i}. Let M be the maximum absolute value of a utility to player i, and let N be the maximum number of actions available to a player. Let $\epsilon > 0$ be arbitrary. Then, if $|x_{-i}(j) - x'_{-i}(j)| < \delta$ for all j, where $\delta = \frac{\epsilon}{MN}$, we can show that $|u_i(\sigma^*, x_{-i}) - u_i(\sigma^*, x'_{-i})| < \epsilon$. This same analysis can be applied to show that our payoff is continuous in the opponent's strategy for many popular distance functions (i.e., for any distance function where one strategy can get arbitrarily close to another as the components get arbitrarily close). For instance this would apply to L1, L2, and earth mover's distance, which have been applied previously to compute distances been strategies for opponent modeling (Ganzfried and Sandholm 2011). Thus, if we are slightly off in our model of the opponent's strategy, even if we are doing a full best response we will do only slightly worse.

4 Exploitation algorithm for Dirichlet prior

As described in Section 1 the Dirichlet distribution is the conjugate prior for the multinomial distribution, and therefore the posterior is also a Dirichlet distribution, with the parameters α_i updated to reflect the new observations. Thus, the mean of the posterior can be computed efficiently by computing the strategy for the opponent in which he plays each strategy in proportion to the updated weight, and Algorithm 1 yields an exact efficient algorithm for computing the Bayesian best response in normal-form games with a Dirichlet prior. However, the algorithm does not apply to games of imperfect information since we do not observe the private information held by the opponent, and therefore do not know which of his action counters we should increment. In this section we will present a new algorithm for this setting. We present it in the context of a representative motivating game where the opponent is dealt a state of private information and then takes publicly-observable action, and present the algorithm for the general setting in Section 4.3.

We are interested in studying the following two-player game setting. Player 1 is given private information state x_i (according to a probability distribution). Then he takes a publicly observable action a_i. Player 2 then takes an action after observing player 1's action (but not his private information), and both players receive a payoff. We are interested

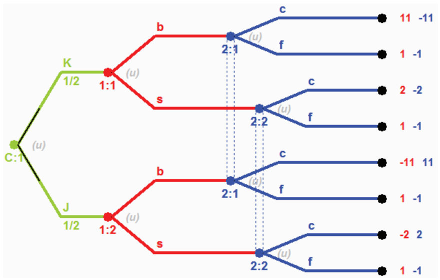


Figure 1: Chance deals player 1 king or jack with probability $\frac{1}{2}$ at the green node. Then player 1 selects big or small bet at a red node. Then player 2 chooses call or fold at a blue node.

in player's 2's problem of inferring the (assumed stationary) strategy of player 1 after repeated observations of the public action taken (but not the private information). Note that this setting is very general. For example, in poker x_i could denote the opponent's private card(s) and a_i denote the amount he bets, and in an ad auction x_i could denote his valuation (e.g., high or low), and a_i could denote the amount he bids (Tang, Wang, and Zhang 2016).

4.1 Motivating game and algorithm

For concreteness and motivation, consider the following poker game instantiation of this setting, where we play the role of player 2. Let's assume that in this two-player game, player 1 is dealt a King (K) and Jack (J) with probability $\frac{1}{2}$, while player 2 is always dealt a Queen. Player 1 is allowed to make a big bet of \$10 (b) or a small bet of \$1 (s), and player 2 is allowed to call or fold. If player 2 folds, then player 1 wins the \$2 pot (for a profit of \$1); if player 1 bets and player 2 calls then the player with the higher card wins the \$2 pot plus the size of the bet.

If we observe player 1's card after each hand, then we can apply the approach described above, where we maintain a counter for player 1 choosing each action with each card that is incremented for the selected action. However, if we do not observe player 1's card after the hand (e.g., if we fold), then we would not know whether to increment the counter for the king or the jack. To simplify analysis, we will assume that we never observe the opponent's private card after the hand (which is not realistic since we would observe his card if he bets and we call); we can assume that we do not observe our payoff either until all game iterations are complete, since that could allow us to draw inferences about the opponent's card. There are no known algorithms even for the simplified case of fully unobservable opponent's private information. We suspect that an algorithm for the case when the opponent's private information is sometimes observed can be constructed based on our algorithm, and we plan to study this problem in future work.

From analysis in the accompanying tech report (Ganzfried and Sun 2016), we are able to compute a closed-form expression for the expectation of the posterior probability that the opponent takes action b with a Jack given that we have just observed him take action b (the other quantities can be computed analogously), which is denoted by $P(b|O,J)$.

$$\frac{B(\alpha_{Kb}+1,\alpha_{Ks})B(\alpha_{Jb}+1,\alpha_{Js})+B(\alpha_{Kb},\alpha_{Ks})B(\alpha_{Jb}+2,\alpha_{Js})}{Z} \tag{1}$$

where the denominator Z is equal to

$$B(\alpha_{Kb}+1,\alpha_{Ks})B(\alpha_{Jb}+1,\alpha_{Js})+B(\alpha_{Kb},\alpha_{Ks})B(\alpha_{Jb}+2,\alpha_{Js})$$

$$+B(\alpha_{Kb}+1,\alpha_{Ks})B(\alpha_{Jb},\alpha_{Js}+1)+B(\alpha_{Kb},\alpha_{Ks})B(\alpha_{Jb}+1,\alpha_{Js}+1).$$

Note that the algorithm we have presented applies for the case where we play one more game iteration and collect one additional observation. However, it is problematic for the general case we are interested in where we play many game iterations, since the posterior distribution is not Dirichlet, and therefore we cannot just apply the same procedure in the next iteration using the computed posterior as the new prior. We will need to derive a new expression for $P(b|O,J)$ for this setting. Suppose that we have observed the opponent play action b for θ_b times and s θ_s times (in addition to the number of fictitious observations reflected in the prior α), though we do not observe his card. Then $P(b|O,J)$ equals

$$\frac{\sum_{i=0}^{\theta_b}\sum_{j=0}^{\theta_s}B(\alpha_{Kb}+i,\alpha_{Ks}+j)B(\alpha_{Jb}+\theta_b-i+1,\alpha_{Js}+\theta_s-j)}{Z} \tag{2}$$

The normalization term is

$$Z=\sum_i\sum_j[B(\alpha_{Kb}+i,\alpha_{Ks}+j)B(\alpha_{Jb}+\theta_b-i+1,\alpha_{Js}+\theta_s-j)$$

$$+B(\alpha_{Kb}+i,\alpha_{Ks}+j)B(\alpha_{Jb}+\theta_b-i,\alpha_{Js}+\theta_s-j+1)].$$

Details of the derivation are in the tech report.

Thus the algorithm for responding to the opponent is the following. We start with the prior counters on each private information-action combination, α_{Kb},α_{Ks}, etc. We keep separate counters θ_b,θ_s for the number of times we have observed each action during play. Then we combine these counters according to Equation 2 in order to compute the strategy for the opponent that is the mean of the posterior given the prior and observations, and we best respond to this strategy, which gives us the same payoff as best responding to the full posterior distribution according to Theorem 1. There are only O(n^2) terms in the expression in Equation 2, so this algorithm is efficient.

4.2 Example

Suppose the prior is that the opponent played b with K 10 times, played s with K 3 times, played b with J 4 times, and played s with J 9 times. Thus $\alpha_{Kb}=10,\alpha_{Ks}=3,\alpha_{Jb}=4,\alpha_{Js}=9$. Now suppose we observe him play b at the next iteration. Applying our algorithm using Equation 1 gives

$$p(b|O,J)=\frac{B(11,3)B(5,9)+B(10,3)(6,9)}{Z}=\frac{2.65209525e^{-7}}{Z}$$

$$p(s|O,J)=\frac{B(11,3)B(4,10)+B(10,3)(5,10)}{Z}=\frac{5.5888056e^{-7}}{Z}$$

$$\longrightarrow p(b|O,J)=\frac{2.65209525e^{-7}}{2.65209525e^{-7}+5.5888056e^{-7}}=0.3218210361.$$

So we think that with a jack he is playing a strategy that bets big with probability 0.322 and small with probability 0.678. Notice that previously we thought his probability of betting big with a jack was $\frac{4}{13} = 0.308$, and had we been in the setting where we always observe his card after gameplay and observed that he had a jack, the posterior probability would be $\frac{5}{14} = 0.357$.

An alternative "naïve" (and incorrect) approach would be to increment α_{Jb} by $\frac{\alpha_{Jb}}{\alpha_{Jb}+\alpha_{Kb}}$, the ratio of the prior probability that he bets big given J to the total prior probability that he bets big. This gives a posterior probability of him betting big with J of $\frac{4+\frac{4}{13}}{14} = 0.308$, which differs significantly from the correct value. It turns out that this approach is actually equivalent to just using the prior:

$$\frac{x + \frac{x}{x+y}}{x+y+1} \cdot \frac{x+y}{x+y} = \frac{x(x+y)+x}{(x+y+1)(x+y)} = \frac{x}{x+y}$$

4.3 Algorithm for general setting

We now consider the general setting where the opponent can have n different states of private information according to an arbitrary distribution π and can take m different actions. Assume he is given private information x_i with probability π_i, for $i = 1, \ldots, n$, and can take action k_i, for $i = 1, \ldots, m$. Assume the prior is Dirichlet with parameters α_{ij} for the number of times action j was played with private information i (so the mean of the prior has the player selecting action k_j at state x_i with probability $\frac{\alpha_{ij}}{\sum_j \alpha_{ij}}$). Assume that action k_{j^*} was observed in a new time step, while the opponent's private information was not observed. We now compute the expectation for the posterior probability that the opponent plays k_{j^*} with private information x_{i^*}.

$$\begin{aligned} & P(A = k_{j^*} | O, C = x_{i^*}) \\ = \quad & \frac{\int \left[q_{k_j^* | x_i^*} \sum_{i=1}^n \left[\pi_i q_{k_{j^*} | x_i} \prod_{h=1}^m \prod_{j=1}^n q_{k_h | x_j}^{\alpha_{jh}-1} \right] \right]}{p(O) \prod_{i=1}^n B(\alpha_{i1}, \ldots, \alpha_{im})} \\ = \quad & \frac{\sum_i \left[\pi_i \prod_j B(\gamma_{1j}, \ldots, \gamma_{nj}) \right]}{Z}, \end{aligned}$$

where $\gamma_{ij} = \alpha_{ij} + 2$ if $i = i^*$ and $j = j^*$, $\gamma_{ij} = \alpha_{ij} + 1$ if $j = j^*$ and $i \neq i^*$, and $\gamma_{ij} = \alpha_{ij}$ otherwise. If we denote the numerator by $\tau_{i^*j^*}$ then $Z = \sum_{i^*} \tau_{i^*j^*}$. Notice that the product is over n terms, and therefore the total number of terms will be exponential in n (it is O($m \cdot 2^n$)).

For the case of multiple observed actions, the posterior is not Dirichlet and cannot be used directly as the prior for the next iteration. Suppose we have observed action k_j θ_j times (in addition to the number of fictitious times indicated by the prior counts α_{ij}). We compute $P(q|O)$ analogously as

$$P(q|O) = \frac{\sum_{i=1}^n \left[\pi_i \sum_{\{\rho_{ab}\}} \prod_{h=1}^m \prod_{j=1}^n q_{k_h | x_j}^{\alpha_{jh}-1+\rho_{jh}} \right]}{p(O) \prod_{i=1}^n B(\alpha_{i1}, \ldots, \alpha_{im})},$$

where the $\sum_{\{\rho_{ab}\}}$ is over all values $0 \leq \rho_{ab} \leq \theta_b$ with $\sum_a \rho_{ab} = \theta_b$ for each b, for $1 \leq a \leq n$, $1 \leq b \leq m$:

$$\sum_{\{\rho_{ab}\}} = \sum_{\rho_{1b}=0}^{\theta_b} \sum_{\rho_{2b}=0}^{\theta_b - \rho_{1b}} \cdots \sum_{\rho_{n-1,b}=0}^{\theta_b - \sum_{r=0}^{n-2} \rho_{rb}} \sum_{\rho_{nb}=\theta_b - \sum_{r=0}^{n-2} \rho_{rb}}^{\theta_b - \sum_{r=0}^{n-1} \rho_{rb}}.$$

The expression for the full posterior distribution is

$$P(q|O) = \frac{\sum_i \left[\pi_i \sum_{\{\rho_{ab}\}} \prod_h B(\alpha_{1h} + \rho_{1h}, \ldots, \alpha_{nh} + \rho_{nh}) \right]}{Z}$$

The total number of terms is $O\left(\left(\frac{(T+n)!}{n!T!} \right)^m \right)$, which is exponential in the number of private information states and actions, but polynomial in the number of iterations.

The following theorem shows an approach for computing products of the beta function that leads to an exponential improvement in the running time of the algorithm for one observation, and reduces the dependence on m for the multiple observation setting from exponential to linear, though the complexity still remains exponential in n and T for the latter. See tech report for full details (Ganzfried and Sun 2016).

Theorem 2. *Define $\gamma_j = \sum_{i=1}^n \gamma_{ij}$ and the empirical probability distribution $\hat{P}_j(i) = \frac{\gamma_{ij}}{\sum_{i=1}^n \gamma_{ij}} = \frac{\gamma_{ij}}{\gamma_j}$. Define the Gamma function $\Gamma(x) = \int_0^\infty x^{z-1} e^{-x} \mathrm{d}x$, for integer x, $\Gamma(x) = (x-1)!$. Now define the entropy of $\hat{P}_i$ as $E(\hat{P}_j) = -\sum_{i=1}^n \hat{P}_j(i) \ln \hat{P}_j(i)$. Then we have $\prod_{j=1}^m B(\gamma_{1j}, \ldots, \gamma_{nj})$ equals*

$$\exp\left(\sum_{j=1}^m \left(-\gamma_j E(\hat{P}_j) - \frac{1}{2}(n-1)\ln(\gamma_j) + \sum_{i=1}^n \ln(P_j(i)) + d \right) \right).$$

Here d is a constant such that $\frac{1}{2}\ln(2\pi)n - 1 \leq d \leq n - \frac{1}{2}\ln(2\pi)$, where $\ln(2\pi) \approx 0.92$.

5 Algorithm for uniform prior distribution

Another prior that has been studied previously is the uniform distribution over a polyhedron. This can model the situation when we think the opponent is playing uniformly within some region of a fixed strategy, such as a specific Nash equilibrium or a "population mean" strategy based on historical data. Prior work has used this model to generate a class of opponents who are more sophisticated than opponents who play uniformly at random over the entire space (Ganzfried and Sandholm 2015)). For example, in rock-paper-scissors, we may think the opponent is playing a strategy uniformly out of strategies that play each action with probability within [0.31,0.35], as opposed to completely random over [0,1].

Let $v_{i,j}$ denote the jth vertex for player i, where vertices correspond to mixed strategies. Let p^0 denote the prior distribution over vertices, where $p^0_{i,j}$ is the probability that player i plays the strategy corresponding to vertex $v_{i,j}$. Let V_i denote the number of vertices for player i. Algorithm 2 computes the Bayesian best response in this setting. Correctness follows straightforwardly by applying Corollary 1 with the formula for the mean of the uniform distribution.

6 Experiments

We ran experiments on the game described in Section 4.1. For the beta function computations we used the Colt Java

Algorithm 2 Algorithm for opponent exploitation with uniform prior distribution over polyhedron

```
Inputs: Prior distribution over vertices $p^0$, response functions $r_t$ for $0 \leq t \leq T$
    $M_0 \leftarrow$ strategy profile assuming opponent $i$ plays each vertex $v_{i,j}$ with probability $p^0_{i,j} = \frac{1}{V_i}$
    $R_0 \leftarrow r_0(M_0)$
    Play according to $R_0$
    for $t = 1$ to $T$ do
        for $i = 1$ to $N$ do
            $a_i \leftarrow$ action taken by player $i$ at time step $t$
            for $j = 1$ to $V_i$ do
                $p^t_{i,j} \leftarrow p^{t-1}_{i,j} \cdot v_{i,j}(a_i)$
            Normalize the $p^t_{i,j}$'s so they sum to 1
        $M_t \leftarrow$ strategy profile assuming opponent $i$ plays each vertex $v_{i,j}$ with probability $p^t_{i,j}$
        $R_t \leftarrow r_t(M_t)$
        Play according to $R_t$
```

math library. For our first set of experiments we tested our basic algorithm which assumes that we observe a single opponent action (Equation 1). We varied the Dirichlet prior parameters to be uniform in $\{1,n\}$ to explore the runtime as a function of the size of the prior (since computing larger values of the Beta function can be challenging). The results (Table 1) show that the computation is very fast even for large n, with running time under 8 microseconds for $n = 500$. However, we also observe frequent numerical instability for large n. The second row shows the percentage of the trials for which the algorithm produced a result of "NaN" (which typically results from dividing zero by zero). This jumps from 0% for $n = 50$ to 8.8% for $n = 100$ to 86.9% for $n = 200$. This is due to instability of algorithms for computing the beta function. We used the best publicly available beta function solver, but perhaps there could be a different solver that leads to better performance in our setting (e.g., it trades off runtime for additional precision). Despite the cases of instability, the results indicate that the algorithm runs extremely fast for hundreds of prior observations, and since it is exact, it is the best algorithm for the settings in which it produces a valid output. Note that $n = 100$ corresponds to 400 prior observations on average since there are four parameters, and that the experiments in previous work used a horizon of 200 hands per match against an opponent (Southey et al. 2005).

n	10	20	50	100	200	500
Time	0.0005	0.0008	0.0018	0.0025	0.0034	0.0076
NaN	0	0	0	0.0883	0.8694	0.9966

Table 1: Results of modifying Dirichlet parameters to be U{1,n} over one million samples. First row is average runtime in milliseconds. Second row is percentage of the trials that output "NaN."

We tested our generalized algorithm for different numbers of observations, using a fixed Dirichlet prior with all parameters equal to as has been done in prior work (Southey et al. 2005). We observe (Table 2) that the algorithm runs quickly for large numbers of observations, though again it runs into numerical instability for large values. As one example, it takes 19 milliseconds for $\theta_b = 101$, $\theta_s = 100$.

n	10	20	50	100	200	500	1000
Time	0.015	0.03	0.36	2.101	10.306	128.165	728.383
NaN	0	0	0	0	0.290	0.880	0.971

Table 2: Results using Dirichlet prior with all parameters equal to 2 and θ_b, θ_s in U{1,n} averaged over 1,000 samples. First row is average runtime (ms), second row is % of trials producing "NaN."

We compared our algorithm against the three heuristics described in previous work (Southey et al. 2005). The first heuristic Bayesian Best Response (BBR) approximates the opponent's strategy by sampling strategies according to the prior and computing the mean of the posterior over these samples, then best-responding to this mean strategy; Max A Posteriori Response heuristic (MAP) samples strategies from the prior, computes the posterior value for these strategies, and plays a best response to the one with highest posterior value; Thompson's Response samples strategies from the prior, computes the posterior values, then samples one strategy for the opponent from these posteriors and plays a best response to it. For all approaches we used a Dirichlet prior with the standard values of 2 for all parameters. For all the sampling approaches we sampled 1,000 strategies from the prior for each opponent and used these strategies for all hands against that opponent (as was done in prior work (Southey et al. 2005)). Note that one can draw samples x_i from a Dirichlet distribution by first drawing independent samples y_i from Gamma distributions each with density Gamma$(\alpha_i, 1) = \frac{y_i^{\alpha_i - 1} e^{-y_i}}{\Gamma(\alpha_i)}$ and then setting $x_i = \frac{y_i}{\sum_j y_j}$.We also tested a best response strategy that knows the actual mixed strategy of the opponent, not just a distribution over his strategies, as well as the Nash equilibrium strategy.[2] Note that the game has a value to us of -0.75, so negative values are not necessarily indicative of "losing."

Table 3 shows that our exact Bayesian best response algorithm (EBBR) outperforms the heuristic approaches, as expected since it is optimal when the opponent's strategy is drawn from the prior. BBR performed best out of the sampling approaches, which is not surprising because it is trying to approximate the optimal approach while the others are optimizing a different objective. All of the sampling approaches outperformed just following the Nash equilibrium, and as expected all exploitation approaches performed worse than playing a best response to the opponent's actual strategy. Note that, against an opponent drawn from a Dirichlet distribution with all parameters equal to 2 and no further observations of his play, our best response would be to always call, which gives us expected payoff of zero. Thus

[2] Note that the Nash equilibrium for player 2 is to call a big bet with probability $\frac{1}{4}$ and a small bet with probability 1 (the equilibrium for player 1 is to always bet big with K and to bet big with probability $\frac{5}{6}$ with J).

for the initial column the actual value for EBBR when averaged over all opponents would be zero. Against this distribution the Nash equilibrium has expected payoff -0.375.

Algorithm	Initial	10	25
EBBR	$\mathbf{0.0003 \pm 0.0009}$	**-0.0024**	**0.0012**
BBR	0.0002 ± 0.0009	-0.0522	-0.138
MAP	-0.2701 ± 0.0008	-0.2848	-0.2984
Thompson	-0.2593 ± 0.0007	-0.2760	-0.3020
FullBR	0.4976 ± 0.0006	0.4956	0.4963
Nash	-0.3750 ± 0.0001	-0.3751	-0.3745

Table 3: Comparison with algorithms from prior work, full best response, and Nash equilibrium using Dirichlet prior with parameters equal to 2. For initial column we sampled ten million opponents from the prior, for 10 rounds we sampled one million, and for 25 rounds 100,000. Results are average winrate per hand over all opponents. Initial column reports 95% confidence interval.

It is interesting that the exploitation approaches (particularly EBBR and BBR) are able to exploit opponents and perform significantly better than the Nash equilibrium strategy just from knowing the prior distribution for the opponents (and without any observations). Previous experiments had also shown that when the sampling approaches are played against opponents drawn from the prior distribution, the winning rates converge, typically very quickly (Southey et al. 2005). For these experiments the performances of all the approaches converged very quickly, and collecting additional observations of the opponent's public action did not seem to lead to an additional improvement. This observation agrees with the findings of the prior results in this setting.

We also tested the effect of using only 10 samples of the opponent's strategy for the sampling approaches. The approaches would then have a noisier estimate of the opponent's strategy, and should achieve lower performance against the actual strategy of the opponent.

Algorithm	Initial	10	25	100
EBBR	$\mathbf{0.000002 \pm 0.0009}$	**0.0019**	**0.0080**	**0.0160**
BBR	-0.1409 ± 0.0008	-0.1415	-0.1396	-0.2254
MAP	-0.2705 ± 0.0007	-0.2704	-0.2660	-0.3001
Thompson	-0.2666 ± 0.0007	-0.2660	-0.2638	-0.3182
FullBR	0.4979 ± 0.0006	0.4980	0.5035	0.5143
Nash	-0.3749 ± 0.0001	-0.3751	-0.3739	-0.3754

Table 4: Comparison of our algorithm with algorithms from prior work (BBR, MAP, Thompson), full best response, and Nash equilibrium using Dirichlet prior with parameters equal to 2. The sampling algorithms each use 10 samples from the opponent's strategy (as opposed to 1000 samples from our earlier analysis). For the initial column we sampled ten million opponents from the prior, for 10 rounds we sampled one million, for 25 rounds 100,000, and for 100 rounds 1,000. Results are average winrate per hand over all opponents. Initial column reports 95% confidence interval.

Thompson and MAP performed very similarly using 10 vs. 1000 samples (these approaches essentially end up selecting a single strategy from the set of samples to be used as the model, and the results indicate that they are relatively insensitive to the number of samples used), but BBR performs significantly worse, achieving payoff around -0.14 with 10 samples vs. payoff close to 0 with 1000 samples. EBBR outperforms BBR much more significantly in this case where BBR uses fewer samples to construct the opponent model. It appears that the sampling approaches actually hurt performance over time when fewer samples are used. BBR, MAP, and Thompson perform clearly worse after 100 game iterations than with fewer iterations, while EBBR performs better as more iterations are used, indicating that it is actually able to perform successful learning in this setting. For the others, the noise from the samples outweighs the gains of learning from additional observations.

7 Conclusion

One of the most fundamental problems in game theory is learning to play optimally against opponents who may make mistakes. We presented the first exact algorithm for performing exploitation in imperfect-information games in the Bayesian setting using the most well-studied prior distribution for this problem, the Dirichlet distribution. Previously an exact algorithm had only been presented for normal-form games, and the best previous algorithm was a heuristic with no guarantees. We demonstrated experimentally that our algorithm can be practical and that it outperforms the best prior approaches, though it can run into numerical stability issues for large numbers of observations.

We presented a general meta-algorithm and new theoretical framework for studying opponent exploitation. Future work can extend our analysis to many important settings. For example, we would like to study the setting when the opponent's private information is only sometimes observed (we expect our approach can be extended easily to this setting) and general sequential games where the agents can take multiple actions (which we expect to be hard, as indicated by the analysis in the tech report). We would also like to extend analysis for any number of agents. Our algorithm is not specialized for two-player zero-sum games (it applies to general-sum games); if we are able to compute the mean of the posterior strategy against multiple opponent agents, then best responding to this strategy profile is just a single agent optimization and can be done in time linear in the size of the game regardless of the number of opponents. While the Dirichlet is the most natural prior for this problem, we would also like to study other important distributions. We presented an algorithm for the uniform prior distribution over a polyhedron, which could model the situation where we think the opponent is playing a strategy from a uniform distribution in a region around a particular strategy, such as a specific equilibrium or a "population mean" based on historical data.

Opponent exploitation is a fundamental problem, and our algorithm and extensions could be applicable to many domains that are modeled as an imperfect-information games. For example, many security game models have imperfect information, e.g., (Letchford and Conitzer 2010; Kiekintveld, Tambe, and Marecki 2010), and opponent exploitation in security games has been a very active area of study, e.g., (Pita et al. 2010; Nguyen et al. 2013). It has also been proposed recently that opponent exploitation can be important in medical treatment (Sandholm 2015).

References

Brown, G. W. 1951. Iterative solutions of games by fictitious play. In Koopmans, T. C., ed., *Activity Analysis of Production and Allocation*. John Wiley & Sons. 374–376.

Chen, X., and Deng, X. 2006. Settling the complexity of 2-player Nash equilibrium. In *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*.

Fudenberg, D., and Levine, D. 1998. *The Theory of Learning in Games*. MIT Press.

Ganzfried, S., and Sandholm, T. 2011. Game theory-based opponent modeling in large imperfect-information games. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Ganzfried, S., and Sandholm, T. 2015. Safe opponent exploitation. *ACM Transactions on Economics and Computation (TEAC)*. Special issue on selected papers from EC-12. Early version appeared in EC-12.

Ganzfried, S., and Sun, Q. 2016. Bayesian opponent exploitation in imperfect-information games. *CoRR* abs/1603.03491.

Johanson, M., and Bowling, M. 2009. Data biased robust counter strategies. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Johanson, M.; Waugh, K.; Bowling, M.; and Zinkevich, M. 2011. Accelerating best response calculation in large extensive games. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Johanson, M.; Zinkevich, M.; and Bowling, M. 2007. Computing robust counter-strategies. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, 1128–1135.

Kiekintveld, C.; Tambe, M.; and Marecki, J. 2010. Robust Bayesian methods for Stackelberg security games (extended abstract). In *Autonomous Agents and Multi-Agent Systems*.

Koller, D.; Megiddo, N.; and von Stengel, B. 1994. Fast algorithms for finding randomized strategies in game trees. In *Proceedings of the 26th ACM Symposium on Theory of Computing (STOC)*, 750–760.

Letchford, J., and Conitzer, V. 2010. Computing optimal strategies to commit to in extensive-form games. In *Proceedings of the ACM Conference on Electronic Commerce (EC)*.

Nguyen, T. H.; Yang, R.; Azaria, A.; Kraus, S.; and Tambe, M. 2013. Analyzing the effectiveness of adversary modeling in security games. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Pita, J.; Jain, M.; Tambe, M.; Ordóñez, F.; and Kraus, S. 2010. Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence Journal* 174(15):1142–1171.

Sandholm, T. 2015. Steering evolution strategically: Computational game theory and opponent exploitation for treatment planning, drug design, and synthetic biology. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. Senior Member Track, Blue Skies Subtrack.

Southey, F.; Bowling, M.; Larson, B.; Piccione, C.; Burch, N.; Billings, D.; and Rayner, C. 2005. Bayes' bluff: Opponent modelling in poker. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 550–558.

Tang, P.; Wang, Z.; and Zhang, X. 2016. Optimal commitments in auctions with incomplete information. In *Proceedings of the ACM Conference on Economics and Computation (EC)*.

Learning Others' Intentional Models in Multi-Agent Settings Using Interactive POMDPs

Yanlin Han, Piotr Gmytrasiewicz
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607

Abstract

Interactive partially observable Markov decision processes (I-POMDPs) provide a principled framework for planning and acting in a partially observable, stochastic and multi-agent environment. It extends POMDPs to multi-agent settings by including models of other agents in the state space and forming a hierarchical belief structure. In order to predict other agents' actions using I-POMDPs, we propose an approach that effectively uses Bayesian inference and sequential Monte Carlo (SMC) sampling to learn others' intentional models which ascribe to them beliefs, preferences and rationality in action selection. Empirical results show that our algorithm accurately learns models of the other agent and has superior performance than other methods. Our approach serves as a generalized Bayesian learning algorithm that learns other agents' beliefs, and transition, observation and reward functions. It also effectively mitigates the belief space complexity due to the nested belief hierarchy.

Introduction

Partially observable Markov decision processes (POMDPs) (Kaelbling, Littman, and Cassandra 1998) provide a principled, decision-theoretic framework for planning under uncertainty in a partially observable, stochastic environment. An autonomous agent operates rationally in such settings by constantly maintaining a belief of the physical state and sequentially choosing the optimal actions that maximize the expected value of future rewards. Although POMDPs can be used in multi-agent settings, it usually treats the effect of other agents' actions as noise and folds it into the state transition function. Examples of such POMDPs are Utile Suffix Memory (McCallum and Ballard 1996), infinite generalized policy representation (Liu, Liao, and Carin 2011), and infinite POMDPs (Doshi-Velez et al. 2015). Therefore, an agent's beliefs about other agents are not part of the solutions of POMDPs.

Interactive POMDPs (I-POMDPs) (Gmytrasiewicz and Doshi 2005) generalize POMDPs to multi-agent settings by replacing POMDP belief spaces with interactive hierarchical belief systems. Specifically, an I-POMDP augments the plain beliefs about the physical states in POMDP by including models of other agents. This augmentation forms a hierarchical belief structure that represents an agent's belief about the physical state, belief about the other agents and their beliefs about others' beliefs, and so on. And such belief structure can be infinitely nested according to its definition. The models of other agents included in the new augmented belief space consist of two types: intentional models and subintentional models. An intentional model ascribes beliefs, preferences, and rationality to other agents (Gmytrasiewicz and Doshi 2005), while a simpler subintentional model, such as finite state controllers (Panella and Gmytrasiewicz 2016), does not. Solutions of I-POMDPs map an agent's belief about the environment and other agents' models to actions. It has been shown (Gmytrasiewicz and Doshi 2005) that the added sophistication of modeling others as rational agents results in a higher value function compared to one obtained from treating others as noise, which implies the modeling superiority of I-POMDPs over other approaches.

However, the interactive belief augmentation of I-POMDPs results in a drastic increase of the belief space complexity, because the agent models grow exponentially as the belief nesting level increases. Therefore, the complexity of the belief representation is proportional to belief dimensions, which is known as the curse of dimensionality. Moreover, since exact solutions to POMDPs are proven to be PSPACE-complete for finite time horizon and undecidable for infinite time horizon (Papadimitriou and Tsitsiklis 1987), the time complexity of more generalized I-POMDPs is at least PSPACE-complete for finite horizon and undecidable for infinite horizon, because an I-POMDP may contain multiple POMDPs or I-POMDPs of other agents. Due to these complexities, a solution which accounts for an agent's belief over an entire intentional model has not been implemented up to date. There are partial solutions that depend on what is known about other agents' beliefs about the physical states (Doshi and Gmytrasiewicz 2009), but they do not include the state of an agent's knowledge about others' reward, transition, and observation functions. Indirect approach such as subintentional finite state controllers (Panella and Gmytrasiewicz 2016) do not include any of these elements either. To unleash the full modeling power of intentional models and mitigate the aforementioned complexities, a robust approximation algorithm is needed. The purpose of this approximation algorithm is to compute the nested interactive belief over all elements of the intentional models and predict other agents' actions. It is crucial to the trade-off between solution quality and computational complexity.

To address this issue, we propose a Bayesian learning method that utilizes customized sequential Monte Carlo sampling (De Freitas, Doucet, and Gordon 2001) to obtain approximate solutions to I-POMDPs and implement the algorithms in a software package.[1] We assume that agents maintain beliefs over intentional models of other agents and make sequential Bayesian updates using observations from the environment. While in multi-agent settings, others agents' models other than their beliefs are usually assumed to be known, in our assumption the modeling agent does not know any information about other agents' transition, observation, reward functions and their beliefs. It only relies on learning indirectly from observations about the environment. Since this Bayesian inference task is analytically intractable due to the need of computing high dimensional integration, we have devised a customized sequential Monte Carlo method starting from the interactive particle filter (I-PF) (Doshi and Gmytrasiewicz 2009). The main idea of this method is to descend the belief hierarchy and sample all model parameters at each nesting level.

Our approach, for the first time, successfully recovers others agents' models over the intentional model space which contains their beliefs, and transition, observation and reward functions. It extends I-POMDP's belief update to larger model space, and therefore it serves as a generalized Bayesian learning method for multi-agent systems in which other agents' beliefs, transition, observation and reward functions are unknown. By approximating Bayesian inference using a customized sequential Monte Carlo sampling method, we significantly mitigate the belief space complexity of I-POMDPs.

The Model

I-POMDP framework

I-POMDPs (Gmytrasiewicz and Doshi 2005) generalize POMDPs (Kaelbling, Littman, and Cassandra 1998) to multi-agent settings by including models of other agents in the belief state space. The resulting hierarchical belief structure represents an agent's belief about the physical state, belief about the other agents and their beliefs about others' beliefs, and can be nested infinitely in this recursive manner. Here we focus on the computable counterparts of infinitely nested I-POMDPs: finitely nested I-POMDPs. For simplicity of presentation, we consider two interacting agents i and j. This formalism generalizes to more number of agents in a straightforward manner.

A finitely nested interactive POMDP of agent i , I-POMDP$_{i,l}$, is defined as:

$$I\text{-}POMDP_{i,l} = \langle IS_{i,l}, A, \Omega_i, T_i, O_i, R_i\rangle \qquad (1)$$

where:

- $IS_{i,l}$ is a set of interactive states, defined as $IS_{i,l} = S \times M_{j,l-1}, l \geq 1$, where S is the set of physical states, $M_{j,l-1}$ is the set of possible models of agent j, and l is the strategy (nesting) level. The set of models, $M_{j,l-1}$, can be divided into two classes, the intentional models, $IM_{j,l-1}$, and subintentional models, $SM_{j,l-1}$. Thus, $M_{j,l-1} = IM_{j,l-1} \cup SM_{j,l-1}$.

  The *intentional* models, $IM_{j,l-1}$, ascribe beliefs, preferences, and rationality in action selection to other agents, thus they are analogous to *types*, θ_j, used in Bayesian games (Harsanyi 1967). The *intentional* models, $\Theta_{j,l-1}$, of agent j at level $l-1$ is defined as $\theta_{j,l-1} = \langle b_{j,l-1}, A, \Omega_j, T_j, O_j, R_j, OC_j\rangle$, where $b_{j,l-1}$ is agent j's belief nested to the level $(l-1)$, $b_{j,l-1} \in \Delta(IS_{j,l-1})$, and OC_j is j's optimality criterion. It can be rewritten as $\theta_{j,l-1} = \langle b_{j,l-1}, \hat{\theta}_j\rangle$, where $\hat{\theta}_j$ includes all elements of the intentional model other than the belief and is called the agent j's frame.

  The *subintentional* models, $SM_{j,l-1}$, constitute the remaining models in $M_{j,l-1}$. Examples of subintentional models are finite state controllers (Panella and Gmytrasiewicz 2016), no-information models (Gmytrasiewicz and Durfee 2000) and fictitious play models (Fudenberg and Levine 1998).

  The $IS_{i,l}$ can be defined in an inductive manner:

$$\begin{aligned}
&IS_{i,0} = S, && \theta_{j,0} = \{\langle b_{j,0}, \hat{\theta}_j\rangle : b_{j,0} \in \Delta(S)\}\\
&IS_{i,1} = S \times \theta_{j,0}, && \theta_{j,1} = \{\langle b_{j,1}, \hat{\theta}_j\rangle : b_{j,1} \in \Delta(IS_{j,1})\}\\
&\ldots\ldots && \qquad (2)\\
&IS_{i,l} = S \times \theta_{j,l-1}, && \theta_{j,l} = \{\langle b_{j,l}, \hat{\theta}_j\rangle : b_{j,l} \in \Delta(IS_{j,l})\}
\end{aligned}$$

- $A = A_i \times A_j$ is the set of joint actions of all agents.
- Ω_i is the set of agent i's possible observations.
- $T_i : S \times A \times S \to [0, 1]$ is the transition function.
- $O_i : S \times A \times \Omega_i \to [0, 1]$ is the observation function.
- $R_i : IS_i \times A \to \mathbb{R}$ is the reward function.

Interactive belief update

Given the definitions above, the interactive belief update can be performed as follows, by considering others' actions and anticipated observations:

$$\begin{aligned}
&b^t_{i,l}(is^t) = Pr(is^t|b^{t-1}_{i,l}, a^{t-1}_i, o^t_i) \qquad (3)\\
&= \alpha \sum_{is^{t-1}} b_{i,l}(is^{t-1}) \sum_{a^{t-1}_j} Pr(a^{t-1}_j|\theta^{t-1}_{j,l-1}) T(s^{t-1}, a^{t-1}, s^t)\times\\
&O_i(s^t, a^{t-1}, o^t_i) \sum_{o^t_j} O_j(s^t, a^{t-1}, o^t_j)\tau(b^{t-1}_{j,l-1}, a^{t-1}_j, o^t_j, b^t_{j,l-1})
\end{aligned}$$

Compared with POMDP, the interactive belief update in I-POMDP takes two additional elements into account. First, the probability of other's actions given his models needs to be computed since the state now depends on both agents' actions (the second summation). Second, the modeling agent needs to update his beliefs based on the anticipation of what observations the other agent might get and how it updates (the third summation).

[1] https://github.com/solohan22/IPOMDP.git

Similarly to POMDPs, the value associated with a belief state in I-POMDPs can be updated using value iteration:

$$V(\theta_{i,l}) = \max_{a_i \in A_i} \Big\{ \sum_{is \in IS} b_{i,l}(is) ER_i(is, a_i) \tag{4}$$
$$+ \gamma \sum_{o_i \in \Omega_i} P(o_i|a_i, b_{i,l}) V(\langle SE_{\theta_i}(b_{i,l}, a_i, o_i), \hat{\theta}_i \rangle) \Big\}$$

where $ER_i(is, a_i) = \sum_{a_j} R_i(is, a_i, a_j) Pr(a_j|\theta_{j,l-1})$.

Then the optimal action, a_i^*, for an infinite horizon criterion with discounting, is part of the set of optimal actions, $OPT(\theta_i)$, for the belief state:

$$OPT(\theta_{i,l}) = \arg\max_{a_i \in A_i} \Big\{ \sum_{is \in IS} b_{i,l}(is) ER_i(is, a_i) \tag{5}$$
$$+ \gamma \sum_{o_i \in \Omega_i} P(o_i|a_i, b_{i,l}) V(\langle SE_{\theta_i}(b_{i,l}, a_i, o_i), \hat{\theta}_i \rangle) \Big\}$$

Sampling Algorithms

The Markov Chain Monte Carlo (MCMC) method (Gilks, Richardson, and Spiegelhalter 1996) is widely used to approximate probability distributions that are difficult to compute directly. Sequential versions of Monte Carlo methods, such as as particle filters (Del Moral 1996), work on sequential inference tasks, especially sequential decision making under Markov assumption. At each time step, a particle filter draws samples (or particles) from a proposal distribution, commonly the conditional distribution $p(x_t|x_{t-1})$ of the current state x_t given the previous x_{t-1}, then uses the observation function $p(y_t|x_t)$ to compute importance weights for all particles and resample them according to the weights.

The Interactive Particle Filter (I-PF) (Doshi and Gmytrasiewicz 2009) was devised as a filtering algorithm for interactive belief update in I-POMDP, which generalizes the classic particle filter algorithm to multi-agent settings. It uses the state transition function as the proposal distribution, which is usually used in a specific particle filter algorithm called bootstrap filter (Gordon, Salmond, and Smith 1993). However, due to the enormous belief space, the I-PF implementation assumes that the other agent's frame $\hat{\theta}_j$ is known to the modeling agent, therefore it simplifies the belief update from $S \times \Theta_{j,l-1}$ to a significantly smaller space $S \times b_{j,l-1}$.

Our interactive belief update described in Algorithm 1 and 2, however, generalizes I-POMDP's belief update to larger intentional model space which contains other agents' beliefs, and transition, observation and reward functions. In the remaining part of this section, we will firstly give a brief introduction of our algorithms and discuss the motivations of each sampling step. Then we will show the major differences between our algorithm and the I-PF, since this generalization is nontrivial. A concrete example of the algorithm is given in Figure 1 in the next section as well.

The Algorithm 1 requires inputs of the modeling agent's prior belief, $\tilde{b}^{t-1}_{k,l}$, which is represented in the form of a set of n samples $is^{(n),t-1}_k$, along with the action, a^{t-1}_k, the observation, o^t_k, and the belief nesting level, $l > 0$. Here k represents either agent i or j, and $-k$ represents the other agent, j or i, correspondingly. We assume that the modeled agent's action set A_{-k}, observation set Ω_{-k} and optimality criteria OC_k are known to all agents. We want to learn the other agent's initial belief about the physical state, b^0_{-k}, the transition function, T_{-k}, the observation function, O_{-k} and the reward function, R_{-k}.

Algorithm 1: Interactive Belief Update

```
$\tilde{b}^t_{k,l}$ = InteractiveBeliefUpdate($\tilde{b}^{t-1}_{k,l}$, $a^{t-1}_k$, $o^t_k$, $l > 0$)
1  For $is^{(n),t-1}_k =< s^{(n),t-1}, \theta^{(n),t-1}_{-k,l-1} >\in \tilde{b}^{t-1}_{k,l}$:
2      sample $a^{t-1}_{-k} \sim P(A_{-k}|\theta^{(n),t-1}_{-k,l-1})$
3      sample $s^{(n),t} \sim T_k(S^t|S^{(n),t-1}, a^{t-1}_k, a^{t-1}_{-k})$
4      for $o^t_{-k} \in \Omega_{-k}$:
5         if $l = 1$:
6             $b^{(n),t}_{-k,0}$ = Level0BeliefUpdate($\theta^{(n),t-1}_{-k,0}$,
                                           $a^{t-1}_{-k}, o^t_{-k}$)
7             $\theta^{(n),t}_{-k,0} =< b^{(n),t}_{-k,0}, \hat{\theta}^{(n),t-1}_{-k,0} >$
8             $is^{(n),t}_k =< s^{(n),t}, \theta^{(n),t}_{-k,0} >$
9         else:
10            $b^{(n),t}_{-k,l-1}$ = InteractiveBeliefUpdate($\tilde{b}^{(n),t-1}_{-k,l-1}$,
                                           $a^{t-1}_{-k}, o^t_{-k}, l-1$)
11            $\theta^{(n),t}_{-k,l-1} =< b^{(n),t}_{-k,l-1}, \hat{\theta}^{(n),t-1}_{-k,l-1} >$
12            $is^{(n),t}_k =< s^{(n),t}, \theta^{(n),t}_{-k,l-1} >$
13            $w^{(n)}_t = O^{(n)}_{-k}(o^t_{-k}|s^{(n),t}, a^{t-1}_k, a^{t-1}_{-k})$
14            $w^{(n)}_t = w^{(n)}_t \times O_k(o^t_k|s^{(n),t}, a^{t-1}_k, a^{t-1}_{-k})$
15            $\tilde{b}^{temp}_{k,l} =< is^{(n),t}_k, w^{(n)}_t >$
16  normalize all $w^{(n)}_t$ so that $\sum_{n=1}^N w^{(n)}_t = 1$
17  resample from $\tilde{b}^{temp}_{k,l}$ according to normalized $w^{(n)}_t$
18  resample $\theta^{(n),t}_{-k,l-1} \sim N(\theta^t_{-k,l-1}|\theta^{(n),t-1}_{-k,l-1}, \Sigma)$
19  return $\tilde{b}^t_{k,l} = is^{(n),t}_k =< s^{(n),t}, \theta^{(n),t}_{-k,l-1} >$
```

The initial belief samples, $is^{(n),t-1}_k$, are generated from the prior nested belief in the similar way as described in the I-PF literature (Doshi and Gmytrasiewicz 2009) except that $T^{(n)}_{-k}, O^{(n)}_{-k}$, and $R^{(n)}_{-k}$ are sampled from their prior distributions as well. Notice that $T^{(n)}_{-k}, O^{(n)}_{-k}$, and $R^{(n)}_{-k}$ are all part of the frame, namely $\hat{\theta}^{(n)}_{-k} =< A_{-k}, \Omega_{-k}, T^{(n)}_{-k}, O^{(n)}_{-k}, R^{(n)}_{-k}, OC_k >$, as appeared in line 7 and 11 in Algorithm 1.

With initial belief samples, the Algorithm 1 starts from propagating each sample forward in time and computing their weights (line 1-15), then it resamples according to the weights and similarity between models (line 16-18). Intuitively, the samples associated with actual observations perceived by agent k will gradually carry larger weights and be resampled more often, therefore they will approximately represent the exact belief. Specifically, for each of

$is_k^{(n),t-1}$, the algorithm samples the other agent's optimal actions a_{-k}^{t-1} from $P(A_{-k}|\theta_{-k}^{(n),t-1})$ (line 2) obtained from POMDP solver Perseus[2] (Spaan and Vlassis 2005). Then it samples the physical state $s^{(n),t}$ using the state transition function $T_k(S^t|S^{(n),t-1}, a_k^{t-1}, a_{-k}^{t-1})$ (line 3). Then for each possible observation, if the current nesting level l is 1, it calls the 0-level belief update, described in Algorithm 2, to update other agents' beliefs over physical state $b_{-k,0}^t$ (line 5 to 8); or it recursively calls itself at a lower level $l-1$ (line 9 to 12), if l is greater than 1. The sample weights $w_t^{(n)}$ are computed according to observation likelihoods of modeling and modeled agents (line 13, 14). Lastly, the algorithm normalizes the weights (line 16), resamples the intermediate particles(line 17) and resamples another time from similar neighboring models using a Gaussian distribution to avoid divergence (line 18).

Algorithm 2: Level-0 Belief Update

```
$b_{k,0}^t$ =Level0BeliefUpdate($\theta_{k,0}^{t-1}, a_k^{t-1}, o_k^t$)
1  get $T_k$ and $O_k$ from $\theta_{k,0}^{t-1}$
2  $P(a_{-k}^{t-1}) = 1/a_{-k}^{t-1}$
3  for $s^t \in S$:
4      for $s^{t-1}$:
5          for $a_{-k}^{t-1} \in A_{-k}$:
6              $P(s^t|s^{t-1}, a_k^{t-1}) =$
                   $T_k(s^t|s^{t-1}, a_k^{t-1}, a_{-k}^{t-1})P(a_{-k}^{t-1})$
7          $sum+ = P(s^t|s^{t-1}, a_k^{t-1})b_{k,0}^{t-1}(s^{t-1})$
8      for $a_{-k}^{t-1} \in A_{-k}$:
9          $P(o_k^t|s^t, a_k^{t-1}) + =$
                   $O_k(o_k^t|s^t, a_k^{t-1}, a_{-k}^{t-1})P(a_{-k}^{t-1})$
10     $b_{k,0}^t = sum \times P(o_k^t|s^t, a_k^{t-1})$
11 normalize and return $b_{k,0}^t$
```

The 0-level belief update, described in Algorithm 2, takes agent model, $\theta_{k,0}^{t-1}$, action, a_k^{t-1}, and observation, o_k^t, as input arguments and returns the belief about the physical state, $b_{k,0}^t$. The other agent's actions are treated as noise (line 2), and transition and observation functions are passed in within the first input argument $\theta_{k,0}^{t-1}$. For each possible action a_{-k}^{t-1}, it computes the actual state transition (line 6) and observation function (line 9) by marginalizing over others' actions, and returns the normalized belief $b_{k,0}^t$. Notice that the transition and observation functions, $T_k(s^t|s^{t-1}, a_k^{t-1}, a_{-k}^{t-1})$ and $O_k(o_k^t|s^t, a_k^{t-1}, a_{-k}^{t-1})$ contained in θ_k^{t-1} , depend on particular model parameters of the actual agent on the 0th level.

Our interactive belief update algorithm differs in three major ways from the I-PF. First, in order to update the belief over intentional model space of other agents, their initial belief, transition function, observation function and reward function in their frames are all unknown and become samples. For instance, the set of n samples of other agents' intentional models $\theta_{-k,l-1}^{(n),t-1} =< b_{-k,l-1}^{(n),t-1}, A_{-k}, \Omega_{-k}, T_{-k}^{(n)}, O_{-k}^{(n)}, R_{-k}^{(n)}, OC_k >$. The observation function of the modeled agents, $O_{-k}^{(n)}(o_{-k}^t|s^{(n),t}, a_k^{t-1}, a_{-k}^{t-1})$ in line 13 of Algorithm 1, is now randomized consequently. Second, the transition and observation functions of the level-0 agent, in line 6 and 9 of Algorithm 2, are passed in as input arguments which correspond to each model sample. Lastly, we add another resampling step in line 18 to avoid divergence, by resampling the model samples from a Gaussian distribution with the mean of current sample value. This additional resampling step is nontrivial, since empirically the samples diverge quickly due to the enormously enlarged sample space.

Experiments

Setup

To demonstrate the correctness of our theoretical framework, we present the results using the multi-agent tiger game (Gmytrasiewicz and Doshi 2005) with various settings. The multi-agent tiger game is a generalization of the classical single agent tiger game (Kaelbling, Littman, and Cassandra 1998). It contains additional observations caused by others' actions, and the transition and reward functions involve others' actions as well.

For the simplicity of presentation, assume there are two agent i and j in the game and the nesting level is 1, then for the two-agent tiger problem: $IS_{i,1} = S \times \theta_{j,0}$, where $S = \{$tiger on the left (TL), tiger on the right (TR)$\}$ and $\theta_{j,0} =< b_j(s), A_j, \Omega_j, T_j, O_j, R_j, OC_j >\}$; $A = A_i \times A_j$ are joint actions of listen (L), open left door (OL) and open right door(OR); Ω_i: $\{$growl from left (GL) or right (GR)$\} \times \{$creak from left (CL), right (CR) or silence (S)$\}$; $T_i = T_j : S \times A_i \times A_j \times S \to [0,1]$; $O_i : S \times A_i \times A_j \times \Omega_i \to [0,1]$; $R_i : IS \times A_i \times A_j \to \mathbb{R}$.

As mentioned before we assume that A_j and Ω_j are known, and OC_j is infinite horizon with discounting. We

Table 1: Parameters for transition, observation and reward functions

S	A	TL	TR
TL	L	p_{T1}	$1-p_{T1}$
TR	L	$1-p_{T1}$	p_{T1}
*	OL	p_{T2}	$1-p_{T2}$
*	OR	$1-p_{T2}$	p_{T2}

S	A	GL	GR
TL	L	p_{O1}	$1-p_{O1}$
TR	L	$1-p_{O1}$	p_{O1}
*	OL	p_{O2}	$1-p_{O2}$
*	OR	$1-p_{O2}$	p_{O2}

S	A	R
*	L	p_{R1}
TL/TR	OL/OR	p_{R2}
TL/TR	OR/OL	p_{R3}

[2] http://www.st.ewi.tudelft.nl/~mtjspaan/pomdp/index_en.html

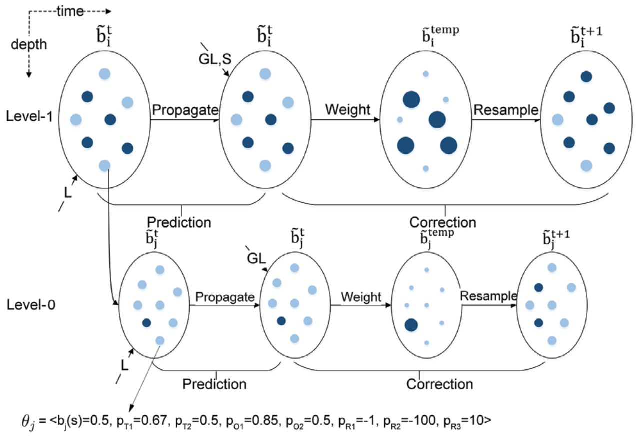


Figure 1: An illustration of interactive belief update algorithm for two agents and level-1 nesting.

want to recover the possible initial belief b_j^0 about the physical state, the transition, T_j, the observation, O_j and the reward, R_j. Thus the main idea of our experiment is to use Bayesian parametric method to parametrize these functions and learn all of them with the help of our sampling algorithm.

The initial belief b_j^0 is a real value between 0 and 1, while the T_j, O_j and R_j can all be parametrized by seven parameters as shown in Table **??**. Thus for the intentional model space, we see that it is a large 8-dimensional parameter space to learn from: $b_j^0 \times p_{T1} \times p_{T2} \times p_{O1} \times p_{O2} \times p_{R1} \times p_{R2} \times p_{R3}$, where $\{b_j, p_{T1}, p_{T2}, p_{O1}, p_{O2}\} \in [0, 1] \subset \mathbb{R}$ and $\{p_{R1}, p_{R2}, p_{R3}\} \in [-\infty, +\infty]$.

Figure 1 illustrates the interactive belief update in the game described above, assuming the sample size is 8. The subscripts denotes the corresponding agents and each dot represents a particular belief sample. The propagating step in implemented in lines 2 to 12 in Algorithm 1, the weighting step is in lines 13 to 16, and the resampling step is in lines 17 and 18. The belief update for a particular level-0 model sample, $\theta_j = \langle 0.5, 0.67, 0.5, 0.85, 0.5, -1, -100, 10\rangle$, is solved using Algorithm 2.

Results

We run experiments with agent j acting according to three different policies shown in Figure 2. And In each experiment, we compare the performance of three different modeling agents: a level-1 I-POMDP, a level-2 I-POMDP and a subintentional model (fictitious play). For brevity we focus on showing results of learning models of the level-1 agent whose policy is in Figure 2 (a), but give an effectiveness comparison among all of them in Figure 5.

To learn all possible models of the agent in Figure 2(a), we assign uninformative prior distributions to each parameter space , which is shown in Figure 3. They are uniform distributions: $\{b_j^0, p_{T1}, p_{T2}, p_{O1}, p_{O2}\} \sim U(0, 1)$, $p_{R1}, p_{R2}, p_{R3} \sim U(-200, 200)$. After 50 time steps, the algorithm converges to a posterior distribution over agent j's intentional models. From the marginal distributions of all

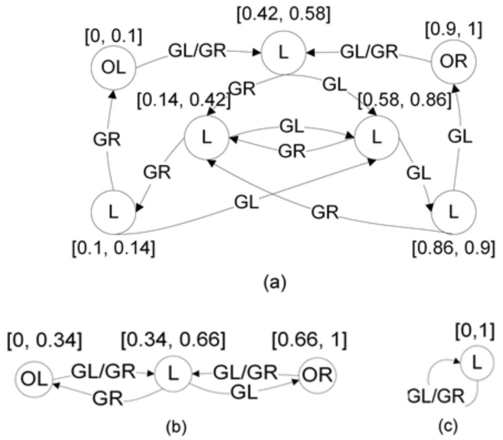


Figure 2: Optimal policies denoted as FSCs of: (a) $\theta_{j1} =< 0.5, 0.67, 0.5, 0.85, 0.5, -1, -100, 10 >$, (b) $\theta_{j2} = \langle$ 0.5, 1, 0.5, 0.95, 0.5, -1, -10, 10 $\rangle$, and (c) $\theta_{j3} = \langle$0.5, 0.66, 0.5, 0.85, 0.5, 10, -100, 10$\rangle$.

parameters, we can see that the majority of samples are centered around the true parameter values.

Due to the limitation of visualizing more than 3D space, here we focus on showing a visualization of the learning process of the level-1 I-POMDP agent. Since the original parameter space is 8-dimensional, we use principal component analysis (PCA) (Abdi and Williams 2010) to reduce it to 2d and plot it out as a 3d histogram, as shown in Figure 4. This time it starts from a slightly informative prior (for the illustrative purpose) and gradually converges to the most likely models. Eventually the mean value of this cluster $\langle$ 0.49, 0.69, 0.49, 0.82, 0.51, -0.95, -99.23, 10.09 $\rangle$ is very close to the actual model. In Figure 5 we show that the learning quality in terms of KL-Divergence, which measures the distance between mean values of the learned model parameters and the ground truth, becomes better as the number of particles increases.

Because agent i is now able to learn others' likely models, he should be capable of predicting j's actions relatively accurately. Therefore, we tested the performance of our algorithm in terms of prediction accuracy towards others' actions, which is the number of incorrect predictions with respect to others' actions over the ground truth. For conciseness, we show the average prediction error rates for the first experiments in Figure 6. We compared the results with other modeling approaches, such as a frequency-based (fictitious play) (Fudenberg and Levine 1998) approach, in which agent j is assumed to choose his action according to a fixed but unknown distribution. The shown results are averaged plots of 10 random runs. It shows that the intentional I-POMDP approaches has significantly lower error rates as agent i perceives more observations, and level-2 I-POMDP performs slightly better than level-1. The frequency based approach has certain learning ability but is far from sophisticated enough to be able to model a rational agent, therefore its performance is worse than both I-POMDP models.

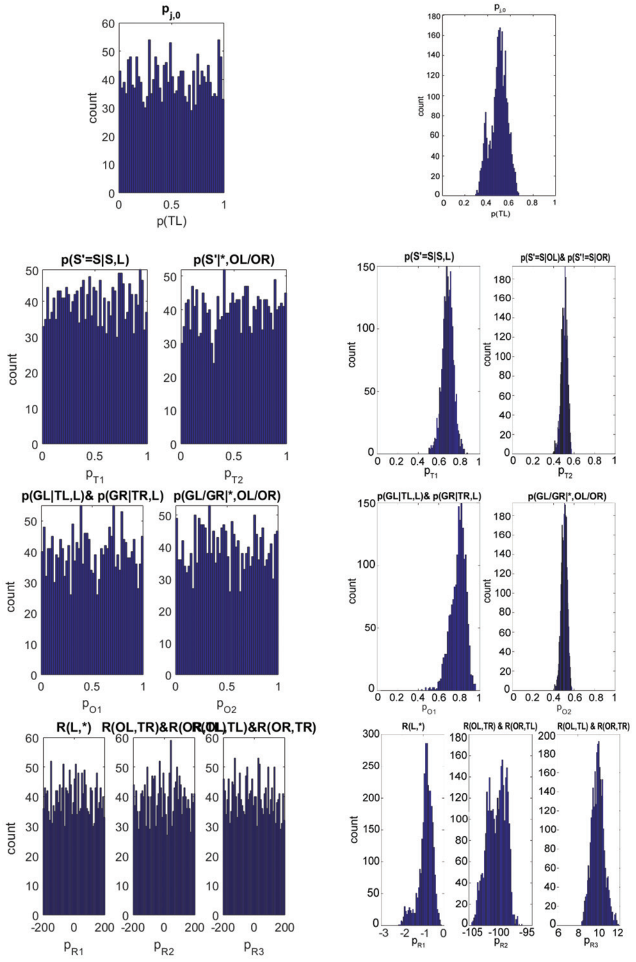


Figure 3: Histograms of assigned uniform priors (left column) and learned posterior distributions (right column) over parameters of model $\theta_{j1} = \langle 0.5, 0.67, 0.5, 0.85, 0.5, -1, -100, 10 \rangle$ in Figure 2(a). The modes of the posteriors are close to the true model parameters.

Conclusions and Future Work

We have described a new approach to learn other agents' intentional models by approximating the interactive belief update using Bayesian inference and Monte Carlo sampling methods. We show the correctness of our theoretical framework using a multi-agent tiger game in which it accurately learns others' models over the entire intentional model space and can be generalized to problems of larger scale in a straightforward manner. Therefore, it provides a generalized Bayesian learning algorithm for multi-agent settings.

For future research opportunities, the applications presented in this paper can be extended to more complicated multi-agent problems. For higher nesting levels, more efforts can be made on leveraging nonparametric Bayesian methods which inherently deal with nested belief structures.

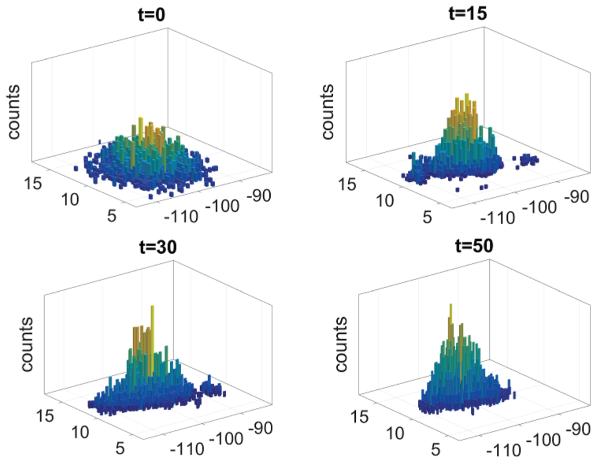


Figure 4: Histogram of all model samples during learning, after projection from 8D to 2D.

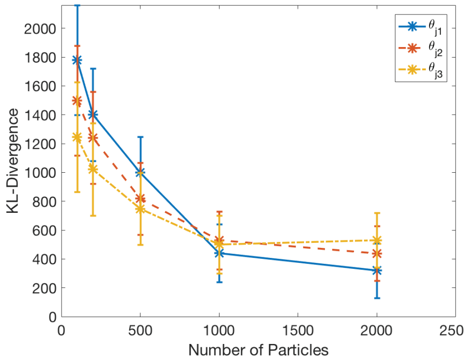


Figure 5: Learning quality measured by KL-Divergence improves as the number of particles increases. It measures the distance of mean values of learned model parameters and the ground truth. The vertical bars are the standard deviations.

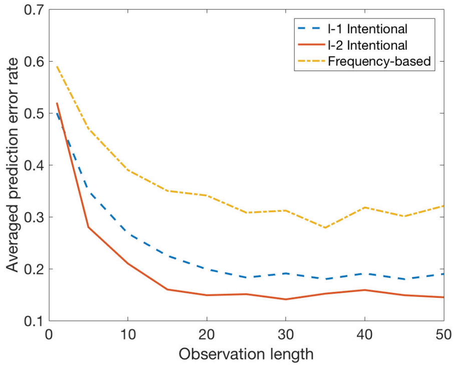


Figure 6: Performance comparisons in terms of prediction error rate vs observation length for $\theta_{j1} = \langle 0.5, 0.67, 0.5, 0.85, 0.5, -1, -100, 10 \rangle$

References

Abdi, H., and Williams, L. J. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational* *statistics* 2(4):433–459.

De Freitas, N.; Doucet, A.; and Gordon, N. 2001. An introduction to sequential monte carlo methods. *SMC Practice. Springer Verlag*.

Del Moral, P. 1996. Non-linear filtering: interacting particle resolution. *Markov processes and related fields* 2(4):555–581.

Doshi, P., and Gmytrasiewicz, P. J. 2009. Monte carlo sampling methods for approximating interactive pomdps. *Journal of Artificial Intelligence Research* 34:297–337.

Doshi-Velez, F.; Pfau, D.; Wood, F.; and Roy, N. 2015. Bayesian nonparametric methods for partially-observable reinforcement learning. *IEEE transactions on pattern analysis and machine intelligence* 37(2):394–407.

Fudenberg, D., and Levine, D. K. 1998. *The theory of learning in games*, volume 2. MIT press.

Gilks, W. R.; Richardson, S.; and Spiegelhalter, D. J. 1996. Introducing markov chain monte carlo. *Markov chain Monte Carlo in practice* 1:19.

Gmytrasiewicz, P. J., and Doshi, P. 2005. A framework for sequential planning in multi-agent settings. *J. Artif. Intell. Res.(JAIR)* 24:49–79.

Gmytrasiewicz, P. J., and Durfee, E. H. 2000. Rational coordination in multi-agent environments. *Autonomous Agents and Multi-Agent Systems* 3(4):319–350.

Gordon, N. J.; Salmond, D. J.; and Smith, A. F. 1993. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE Proceedings F (Radar and Signal Processing)*, volume 140, 107–113. IET.

Harsanyi, J. C. 1967. Games with incomplete information played by bayesian players, i–iii part i. the basic model. *Management science* 14(3):159–182.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and acting in partially observable stochastic domains. *Artificial intelligence* 101(1):99–134.

Liu, M.; Liao, X.; and Carin, L. 2011. The infinite regionalized policy representation. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 769–776.

McCallum, A. K., and Ballard, D. 1996. *Reinforcement learning with selective perception and hidden state*. Ph.D. Dissertation, University of Rochester. Dept. of Computer Science.

Panella, A., and Gmytrasiewicz, P. J. 2016. Bayesian learning of other agents' finite controllers for interactive pomdps. In *AAAI*, 2530–2536.

Papadimitriou, C. H., and Tsitsiklis, J. N. 1987. The complexity of markov decision processes. *Mathematics of operations research* 12(3):441–450.

Spaan, M. T., and Vlassis, N. 2005. Perseus: Randomized point-based value iteration for pomdps. *Journal of artificial intelligence research* 24:195–220.

Model-Based Reinforcement Learning under Periodical Observability

Richard Klima,[1] Karl Tuyls,[1] Frans A. Oliehoek[1]
[1]University of Liverpool, United Kingdom
{richard.klima, k.tuyls, frans.oliehoek}@liverpool.ac.uk

Abstract

The uncertainty induced by unknown attacker locations is one of the problems in deploying AI methods to security domains. We study a model with partial observability of the attacker location and propose a novel reinforcement learning method using partial information about attacker behaviour coming from the system. This method is based on deriving beliefs about underlying states using Bayesian inference. These beliefs are then used in the QMDP algorithm. We particularly design the algorithm for spatial security games, where the defender faces intelligent and adversarial opponents.

Introduction and Motivation

In security domains we often face several uncertainties which make acting effectively very difficult. Overcoming the uncertainties is one of the main challenges in order to deploy AI techniques in real-world applications. The reasoning agent has often an access to extra information about the environment which if used properly can help significantly in effective strategy-making. In security games this knowledge can come from several types of surveillance available to the agent. We focus on a model-based approach, where we continually learn and improve our knowledge about the opponent behaviour. The main uncertainty lies in not being able to always observe the opponent location. To tackle this challenge we develop a statistical probability model to enable us to reason about opponent location. We base opponent location modelling on observed frequencies of transition tuples and prior information about the environment e.g. target location. Our proposed algorithm is based on the QMDP (Littman, Cassandra, and Kaelbling 1995) algorithm, which combines the standard Q-learning with belief states in partially observable domains. We extend this algorithm with Bayesian inference update using prior information about the environment.

We describe our work in terms of a taxonomy proposed in (Hernandez-Leal et al. 2017), where the authors discuss a classification in terms of environment observability, opponent adaptation capabilities and how the agent deals with non-stationarity. We assume observability of the agent's local reward and partial observability of opponent's actions. The opponent is assumed to adapt his strategy within some bounds, thus we restrict his behaviour from abrupt/drastic changes. This is explained by the concept of bounded rationality, which is often used in security games (Pita et al. 2010). Such a concept allows us to learn a model of opponent behaviour and use it to form the defender strategy.

This paper is motivated by the domain of Green Security Games (Fang, Stone, and Tambe 2015), with a focus on the problem of Illegal Rhino Poaching (Montesh 2013) and on ways how to learn effective ranger strategies in order to mitigate rhinos killings. Nevertheless, our proposed method is applicable to other spatial security game scenarios which can be modelled on a grid (graph). The problem belongs to a domain of pursuit-evasion games. There has been a lot of work on computing exact solutions and describing their theoretical properties in security games, mostly using the equilibria concepts e.g. Nash equilibria or Stackelberg equilibria (Korzhyk et al. 2011). This line of research has been important as a theoretical underpinning of the field, however, these methods are often difficult to deploy in real world settings due to some strict assumptions or severe simplifications. A different approach from computing exact solution strategy is to learn the strategy from interacting with the environment. This approach helps to overcome some of the assumptions of the theoretical approaches.

The domain of security games can be modelled as a reward-based system, where the agents obtain rewards and thus can learn strategies. The problem can be approached by Multi-agent Reinforcement Learning (MARL) using the Markov Decision Process (MDP) framework. In MARL it is very difficult to learn optimal strategies because of the *moving target* problem (Tuyls and Weiss 2012), where all agents are assumed to be adapting to each others behaviour. In security games we face an additional complexity caused by the uncertainty about the attacker, who can be intelligent and strategic. One of the possible uncertainties about the attacker is his location, which might not be observable or only partially observable. We focus on a special case of partial (limited) observability which is inspired by the board game *Scotland Yard* where the player gets to observe the opponent location only periodically e.g. every 3 time steps. We claim that this type of observability is quite common in security domains where the defender gets to observe an opponent location by obtaining some extra information. For instance in the green security game scenarios like Rhino Poaching problem, the rangers can be informed by the villagers living nearby about the current location of the poachers, or this information can also come from surveillance by drones (Montesh 2013). In our model we assume an adversarial adaptive opponent who might be able to observe the defender behaviour. Our main goal is to make use of the extra information about the attacker location in reinforcement learning, obtaining an adaptive strategy to apprehend the attacker.

Related Work

This paper is situated in the field of Multi-agent Reinforcement Learning (MARL), which is a very active field of research since there has been substantially less work done in MARL compared to single-agent RL due to the increased complexity. For more information on MARL we refer the reader to surveys (Bloembergen et al. 2015) or (Hernandez-Leal et al. 2017). We divide this section into several fields of research, which are closely related to this paper. These consist of Partially Observable Markov Decision Processes (POMDP), Bayesian Reinforcement Learning and Security Games (SG). We state the related work respectively.

Partially observable problems are often modelled as Partially Observable Markov Decision Processes (POMDP) (Kaelbling, Littman, and Cassandra 1998). Related to our work is algorithm BA-POMDP proposed in (Ross et al. 2007), where the authors combine Bayesian approach with POMDP model or the learning version BA-POMCP (Katt, Oliehoek, and Amato 2017). We also mention Bayesian Q-learning proposed in (Dearden, Friedman, and Russell 1998), which uses Bayesian inference combined with Q-learning to model the value function. The domain of Bayesian learning can be divided into probabilistic modelling of transition function, value function, reward function or policy. In this paper we focus on probabilistic modelling of transition function. We also propose a combination of Bayesian approach and Q-learning, however in substantially different way. Our method uses Bayesian approach to model transition function to derive belief states, modelling the partially observable attacker behaviour.

Security games have gained a lot of attention in recent years due to their successful application on real-world security threats. Examples include the ARMOR system for airport security (Pita et al. 2008) or the PROTECT system for scheduling Coast Guard (Shieh et al. 2012). Additionally some work has focused on Green Security Games for poaching problems (Fang, Stone, and Tambe 2015) or Border Patrol (Klima, Lisy, and Kiekintveld 2015). Some of these security games however, do not consider space or time, i.e. the time it takes the defender to travel to the target node, as part of the model. Recently, reinforcement learning has been applied to spatial security games (Klima, Tuyls, and Oliehoek 2016) to tackle the spatial component. Spatial security games are also often modelled as extensive form games (Korzhyk et al. 2011). There has been lot of work in computing the optimal strategies online or offline, especially for zero-sum games (Bosansky et al. 2016), (Jain et al. 2011). We also mention the work of (An et al. 2012), which computes the optimal defender strategy to a learning attacker who can only partially observe the defender and updates his beliefs using Dirichlet distribution. In this paper we assume the attacker can fully observe the defender past moves and plays fictitious play (Fudenberg and Levine 1996). We address this by learning the defender strategy. Fictitious play is well-defined in 1D space, but it is more complicated in 2D space. Recently, (Heinrich, Lanctot, and Silver 2015) showed the extension of fictitious play into extensive form games implemented in behavioural strategies with similar properties as the original fictitious play.

Model

We study the problem of effective decision making in spatial security games. Our focus is a spatial security game played on a graph with two non-cooperative players with opposing (not strictly, assuming general-sum game) goals. We define these two players as the defender and the attacker. In this work we use the terms defender/agent and attacker/opponent interchangeably. The model is inspired by the Green Security Game framework where we are interested in the problem of Illegal Rhino Poaching. In such a problem the rangers (the defender) tries to apprehend illegal rhino poachers (the attacker) and thus protect the rhinos (targets) from being poached. The environment is a wildlife reservation, which can be modelled as a graph (grid).

We define this framework in terms of Stochastic game (Shapley 1953) using Markov Decision Process (MDP) model. A state is defined as a combination of locations of the defender and the attacker in the grid, an action is defined for the defender as moving from one place in the grid to another and a reward is defined as a positive signal for apprehending the attacker. A Stochastic (Markov) game as described in (Wiering and van Otterlo 2013) chapter 14.3.1. is defined as a tuple $(n, S, A_1 \dots A_n, R_1 \dots R_n, T)$ where n is number of agents in the system, S is a finite set of system states, A_k is the action set of agent k, $R_k : S \times A_1 \times ... \times A_n \rightarrow \mathbb{R}$ is the reward function of agent k and $T : S \times A_1 \times ... \times A_n \times S \rightarrow [0, 1]$ is the transition function.

Observability in spatial security game

In our security game we assume that the defender can always observe his own location but sometimes cannot observe the attacker location, thus cannot fully observe the underlying state. Agent's observations consist of either full observation of the state or an observation of only own location. Therefore, the defender needs to maintain beliefs $b(s)$ over states which give him the probability of being in a state s. In every time step we restrict the set of possible states by (i) physical structure of the map (gridworld) and (ii) by observations of attacker location in previous time steps. We use the notion of *information set* from extensive-form game theory to denote such restricted set of possible states. We define such a restricted state space as a subset of the original state space denoted $\bar{S} \subseteq S$.

In Figure 1 we show an example of a small grid world and corresponding extensive-form tree with information set. The

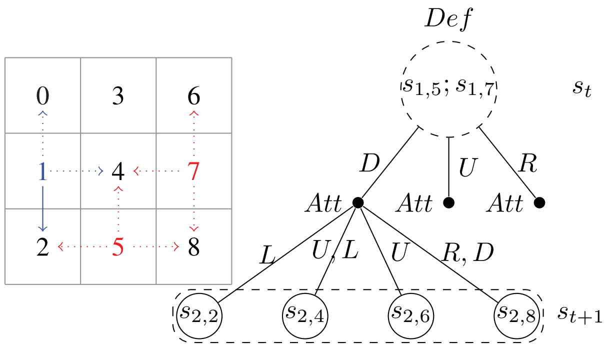


Figure 1: Example of states in information sets for the case where neither the current state nor the succeeding state is observed. We need to reason over two information sets at time t and $t+1$. The defender is in location 1 and chooses action *down* (D), the attacker is in location 5 or 7.

defender is unsure about the state, it is either $s_{1,5}$ or $s_{1,7}$, because the defender can always observe his own location (tile 1) but might be unsure about the attacker location (either tile 5 or tile 7). The figure captures a decision point, where the defender decides to go *down* (D). The defender reasons about the possible attacker actions and about the resulting attacker location in order to form the information set. We update the beliefs only over the states in given information set.

We study a scenario with a periodical observability i.e. the defender gets to observe the attacker location every k steps. This type of observability is inspired by the board game *Scotland Yard*. We compare this type of observability with a full observability of the attacker location and partial observability i.e. knowing only agent's own position.

Attacker behaviour model

In security domains we often face an adversarial opponent who is potentially intelligent and can observe the defender behaviour to some extent and plan his strategy accordingly. In our model the attacker plays a version of fictitious play (Fudenberg and Levine 1996), considering an intelligent and adaptive opponent. We assume that the attacker can observe all the past moves of the defender. This assumption is rather strong but describes the worst-case scenario in security games. We also choose the fictitious play because of its properties. It is a best response to defender past moves and is guaranteed to converge to Nash equilibrium in some games (e.g. zero-sum games (Robinson 1951)).

Statistical approach to uncertainty

We assume that both players know the environmental model i.e. state space, action space and reward function. However the defender is uncertain about the location of the opponent and his strategy. Our main goal is to act efficiently under this uncertainty. In security games the defender has often access to some extra information about the attacker whereabouts, which we use to deal with this uncertainty.

We define a discrete random variable X in the restricted space $\bar{S}'$ of the succeeding states given by the information set. Thus, we have a discrete probability distribution of the succeeding states $P(X = s') : \forall s' \in \bar{S}'$ parametrized by a vector θ, where $\sum_i^k \theta_i = 1$ and $P(X = s'|\theta) = \theta_i$. We assume that the defender can observe some of the transitions defined by a transition tuple (s, a, s'). The defender stores these transitions and form a vector $\Phi = (\phi_1, \ldots, \phi_k)$ of transition occurrences; for example $\phi_{s'}^{sa}$ is a number of past observations of a transition from state s taking action a to state s'.[1] The defender in our model forms beliefs about the possible states defined by the information set. The information set is build based either on a direct observation or on reasoning about previous attacker locations (see Figure 1). These beliefs are probabilities defined by the vector θ, e.g. θ_s is the probability of being in state s. The goal is to derive these probabilities given the past observed transitions, thus we need to compute the probability distribution $P(\theta|\Phi)$. Note that the total number of observations (of the succeeding states for given state and action) is $n = \sum_{\bar{S}'} \phi_i^{sa}$, note that $|\bar{S}'|$ is the size of the information set.

Firstly, we assume that probability distribution $P(\Phi|\theta)$ follows a multinomial distribution with parameters n and θ. Thus, we can write the probability mass function of multinomial distribution as:

$$P(\Phi|\theta) \sim f(\Phi|n,\theta) = \frac{n!}{\prod_{\bar{S}'} \phi_i^{sa}!} \prod_{\bar{S}'} \theta_i^{\phi_i^{sa}} \quad (1)$$

Note that $\frac{n!}{\prod_{\bar{S}'} \phi_i^{sa}!}$ is the total number of possible observation sequences giving the vector Φ.

The defender assumably has prior knowledge about the environment e.g. target location, which we use as a prior for Bayesian inference. We define the prior probability as a Dirichlet distribution $Dir(\alpha)$, which is defined for hyperparameters α. $Dir(\alpha)$ is a probability distribution over parameters θ of multinomial distribution and is also its conjugate prior. The hyperparameters α can be seen as pseudo-observations to complement the actual observed transitions i.e. the transition counts Φ. $Dir(\alpha)$ is defined using Γ function as:

$$Dir(\theta|\alpha) = \frac{\Gamma(\sum_i^k \alpha_i)}{\prod_i^k \Gamma(\alpha_i)} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1} \quad (2)$$

We already defined the likelihood as multinomial distribution using the transition counts Φ (see Equation 1) and thus, we can write the posterior using Bayes' rule as:

$$Dir(\theta|\Phi) \propto Multi(\Phi|n,\theta) Dir(\alpha) \quad (3)$$

We can then write $P(\theta|\Phi) = Dir(\Phi + \alpha)$. In this work we are not interested in the full posterior $P(\theta|\Phi)$, because we want only a point estimate to determine a belief about states of the model. We focus on the expected value of the distribution to obtain the belief given past observations of the transitions and prior information. The expected value of

[1] For transition counts we use notation ϕ following the previous work e.g. (Ross et al. 2007).

the posterior distribution is defined for multinomial likelihood and Dirichlet prior in Bayes rule as $E_{Dir(\Phi+\alpha)}[\theta_i] = \frac{\phi_i^{sa}+\alpha_i}{n+\sum_{j=1}^{k}\alpha_j}$. Note that when deriving point estimates of posterior distribution we do not need marginal distribution of data (normalizing constant) $P(\Phi)$.

We can now obtain the belief $b^{oa}(s')$ about the succeeding state s' given the observation o and action a. The observation gives us belief $b(s)$ about the state s, transition counts $\phi_{s'}^{sa}$ and priors α as:

$$\begin{aligned} b^{oa}(s') = \sum_{s\in\bar{S}} b(s) E_{Dir(\Phi+\alpha)}[\theta_{s'}] = \\ \sum_{s\in\bar{S}} b(s) \frac{\phi_{s'}^{sa} + \alpha_{s'}}{n_{s\cdot}^{a} + \sum_{j\in\bar{S}'}\alpha_j} \end{aligned} \tag{4}$$

where $n_{s\cdot}^{a}$ is the sum of all the observations for given state s and action a.

We now discuss the setting of the hyperparameters α. We believe that the attacker behaviour is steered by the location of the targets which is known information to both of the players at the beginning of the game. Therefore, prior for each node (location) is defined as $\alpha_{node} = \frac{1}{SP(node,target)+1} * priorConfidence$, where $SP(node, target)$ is the shortest path to the nearest target from the given node, $priorConfidence$ depends on number of observations and potentially other influences determining the confidence in comparison to actual observations. Note that the prior is defined for a location of the attacker ignoring the location of the defender. This simplification comes from the assumption that the attacker cannot fully observe the defender location in given game episode (but knows the past moves) and is mainly steered by location of the targets.

Saving transition counts in partial observability

The defender uses a model-based learning approach. In each time step he saves a transition tuple observed in the current transition. In the case he cannot fully observe the current or/and the succeeding state he updates the transition counts $\phi_{s'}^{sa}$ proportionally to the beliefs $\phi_{s'}^{sa}$ += $b(s)b(s')$. Therefore, the stronger the belief about a particular state is the more he updates the corresponding value in the vector Φ. Note that for fully observed states s and s' the update is equal to 1.

Q-learning with Bayesian Inference

We combine the inference of probabilities of different states in given information set with standard temporal difference learning algorithm TD(0) i.e. Q-learning, where we use QMDP algorithm (Littman, Cassandra, and Kaelbling 1995). We present BayesQMDP in Algorithm 1. The action-selection on line 4 is ϵ-greedy proportional to the belief, meaning that the action a from state s is more likely to be chosen with increasing probability of being in the state s and increasing Q-value for that state and action. On line 5 we update Q-values using the belief about states $b(s)$. The learning rate λ is linked to the belief we have about the state; the less certainty (lower probability) about being in the state the less we update the Q-value and vice-versa (smaller learning rate).[2] The value function on line 6 is a sum over maximal Q-values of the succeeding states weighted by the probability (belief) of going to those states. The belief update on line 7 uses the expected value of the posterior probability distribution as explained in Equation 4. Finally, on line 8 we update the transition count vector $\phi_{s'}^{sa}$.

Algorithm 1 BayesQMDP

```
1: **Input:** priors $\alpha$, parameters $\lambda$, $\gamma$
2: **Init:** $s_0, Q(s,a) = 0,\ \phi_{s'}^{sa} = 0\, \forall s, s' \in S\, \forall a \in A$
3: **for** t in game **do**
4:     $\epsilon$-greedy: $a = \arg\max_a \sum_{\bar{S}} b(s) * Q(s,a)$
5:     $\forall s$: $Q(s,a) = (1 - b(s)\lambda)Q(s,a) + b(s)\lambda(r + \gamma V(s'))$
6:     where $V(s') = \sum_{\bar{S}'} b(s') \max_a Q(s',a)$
7:     $b^{oa}(s') = \sum_{\bar{S}} b(s) \frac{\phi_{s'}^{sa} + \alpha_{s'}}{n_{s\cdot}^{a} + \sum_{\bar{S}'} \alpha_j}$
8:     $\phi_{s'}^{sa}$ += $b(s) * b(s')$
```

Experiments

In this section we compare the proposed BayesQMDP with two baseline algorithms based on standard Q-learning. We show two different gridworlds.

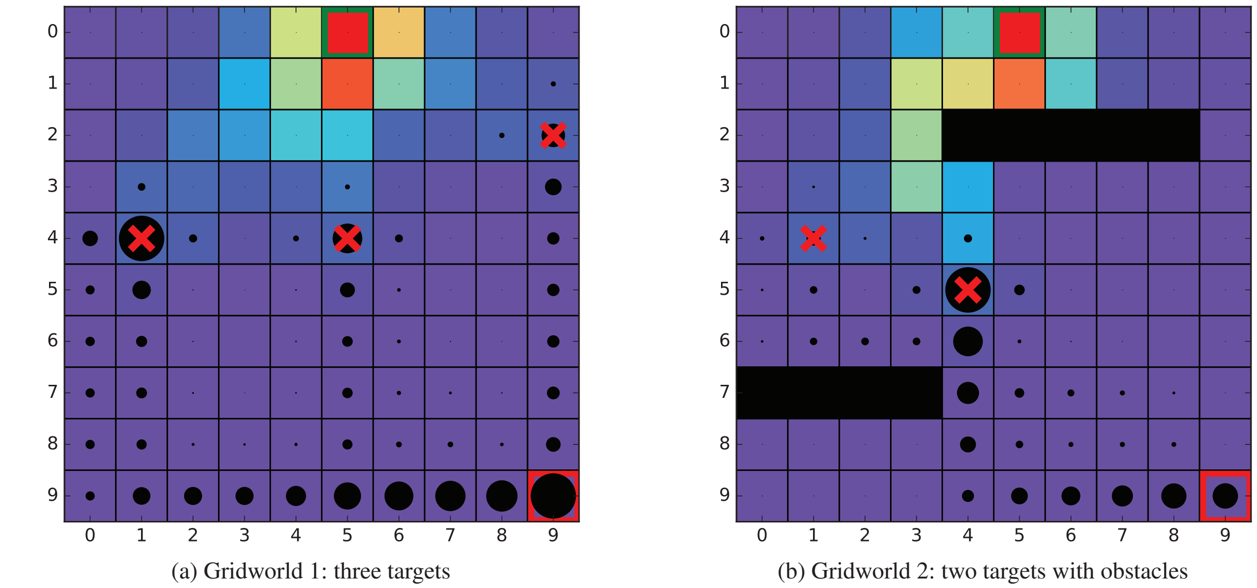
Security game gridworld

We perform the experiments on a grid of size 10x10, thus the state space has size 100^2 i.e. 100 possible locations for each player (as explained before a state is defined by the location of the defender and the attacker). The defender starts on top in the middle and the attacker starts in the right bottom corner. In our model the attacker chooses a best response to defender past locations in the grid world, which is the shortest path from start node to target location weighted by defender visits in each node over all the targets. The attacker chooses his path at the beginning of every episode. Every target has some probability p of success; for example once the poacher (attacker) reaches the target, he has p probability of poaching a rhino in which case the game ends. If the attacker gets to a target (e.g. area with a rhino) and is not successful, he makes a random move from the target node and tries again in the next time step.

We present experiments with two and three targets, each with probability $p = 0.3$ of successful attack. If the defender is in the same location as the attacker, the attacker is apprehended and the defender receives a positive reward. If the defender apprehends the attacker or the attacker successfully attacks a target, the game (episode) ends. As a performance metric we use the percentage of defender wins i.e. the percentage of attacker apprehensions.

In our experiments we compare the proposed algorithm BayesQMDP with two baselines. The first baseline is a standard Q-learning with full observability of the attacker. The second baseline is also a standard Q-learning but this time

[2] For learning rate we use λ instead of the common notation α to distinguish from the hyperparameter.



(a) Gridworld 1: three targets

(b) Gridworld 2: two targets with obstacles

Figure 2: 10x10 gridworlds, targets depicted by red crosses, defender starts at position [0,5] (green square), attacker starts at position [9,9] (red square). The heatmap shows defender visits in each tile and the black dots show attacker visits in each tile (size of the black dots represents the number of visits).

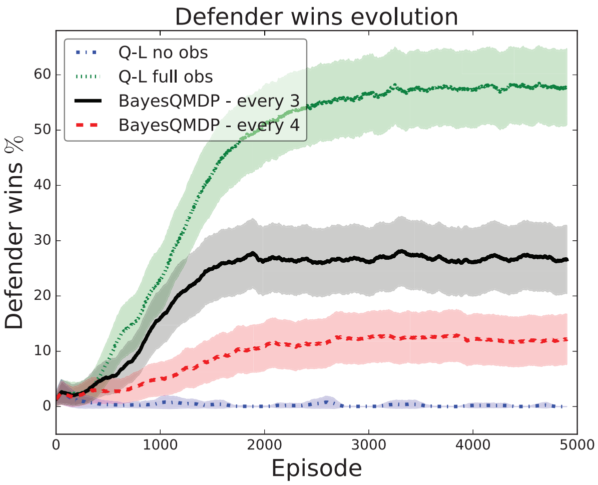


Figure 3: Defender wins for BayesQMDP - Gridworld 1

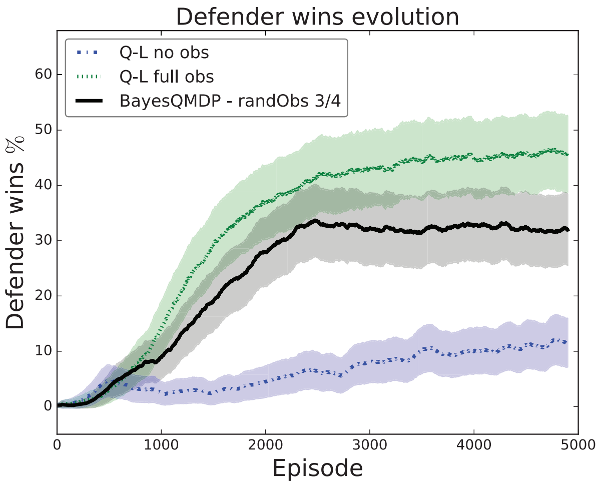


Figure 4: Defender wins for BayesQMDP - Gridworld 2

with no observability of the attacker. In this method the state is defined as the defender location only i.e. ignoring the attacker. All the algorithms use standard settings of learning rate $\lambda = 0.05$, discount factor $\gamma = 0.99$ and fading exploration rate $\epsilon = 0.01 + \frac{0.99}{e^{0.001t}}$. We experiment with different number of periodical observability steps. We use two different gridworlds; Gridworld 1, which has three targets and no obstacles and Gridworld 2, which has two targets and some obstacles. See Figure 2a showing Gridworld 1, the green and red hollow rectangles show the players starting nodes - defender and attacker respectively. The red crosses represent the targets. The heatmap shows defender visits in every node and the black dots show attacker visits (the bigger the dot the more often the attacker was in that node). The gridworlds are shown for one of the baselines - Q-learning with full observability.

In Figure 3 we show the performance of BayesQMDP against the baseline algorithms in GridWorld 1 (Figure 2a) with 95% confidence intervals. The black solid curve is for the case where the defender gets to observe the attacker location every 3rd time step and the red dashed curve is observing every 4th time step. The full observability Q-learning (the green dotted curve) performs the best which is expected, however the no observability Q-learning (the dash-dot blue

Grid	*Q-L full obs*	*Q-L no obs*	*BayesQMDP*
1	$58.0\%, \pm 6.82\%$	$0\%, \pm 0\%$	$26.7\%, \pm 6.12\%$
2	$45.8\%, \pm 6.89\%$	$11.6\%, \pm 4.43\%$	$32.2\%, \pm 6.47\%$

Table 1: Average wins over last 100 episodes with 95% confidence intervals

curve) gets exploited by the attacker's fictitious play. The BayesQMDP algorithm gives us a good performance in the partial observability. Note that observing the attacker every 4th time step can lead to information set size of 25 states in the worst case (4 actions in every state - without repeating the same states).

In Figure 4 there is BayesQMDP compared to the two baselines for Gridworld 2 (Figure 2b). In this experiment we do not assume fixed number of steps to observe the attacker location, instead we sample uniform at random between observing the attacker every 3rd and every 4th time steps to account for any potential synchronisation. One can observe that BayesQMDP gives superior performance compared to no observability case and is close to full observability case. This result shows the effective behaviour of BayesQMDP in partial observability.

Every experiment is run 200 times with 5000 episodes each and averaged over to get significant results. In Table 1 we show the defender wins in the last 100 episodes for all the compared algorithms, we also state 95% confidence intervals. Note that for BayesQMDP we state the results for observability every 3rd time step for Gridworld 1 and random observability between 3rd and 4th step for Gridworld 2.

Conclusion

We have proposed a new algorithm combining QMDP and Bayesian inference called BayesQMDP, which can effectively use partial information about attacker location. We compared this algorithm with two very simple baseline algorithms to demonstrate the initial performance and promising behaviour. The algorithm is experimentally shown to converge against our version of fictitious play. This is a preliminary experimental evaluation of BayesQMDP and we leave further analysis of the proposed algorithm for future work. The next step is comparing BayesQMDP to stronger baseline algorithms such as BA-POMCP (Katt, Oliehoek, and Amato 2017) or DRQN (Hausknecht and Stone 2015).

References

An, B.; Kempe, D.; Kiekintveld, C.; Shieh, E.; Singh, S.; Tambe, M.; and Vorobeychik, Y. 2012. Security Games with Limited Surveillance. In *AAAI Conference on Artificial Intelligence*, 1241–1248.

Bloembergen, D.; Tuyls, K.; Hennes, D.; and Kaisers, M. 2015. Evolutionary Dynamics of Multi-agent Learning: A Survey. *Journal of Artificial Intelligence Research* 53:659–697.

Bosansky, B.; Lisy, V.; Lanctot, M.; Cermak, J.; and Winands, M. H. M. 2016. Algorithms for Computing Strategies in Two-player Simultaneous Move Games. *Artificial Intelligence* 237:1–40.

Dearden, R.; Friedman, N.; and Russell, S. 1998. Bayesian Q-learning. *American Association of Artificial Intelligence (AAAI)* 761–768.

Fang, F.; Stone, P.; and Tambe, M. 2015. When Security Games Go Green: Designing Defender Strategies to Prevent Poaching and Illegal Fishing. *International Joint Conference on Artificial Intelligence* 2589–2595.

Fudenberg, D., and Levine, D. 1996. *The Theory of Learning in Games*. The MIT Press.

Hausknecht, M., and Stone, P. 2015. Deep Recurrent Q-Learning for Partially Observable MDPs. *arXiv preprint arXiv:1507.06527*.

Heinrich, J.; Lanctot, M.; and Silver, D. 2015. Fictitious Self-Play in Extensive-Form Games. In *International Conference on Machine Learning*.

Hernandez-Leal, P.; Kaisers, M.; Baarslag, T.; and Munoz de Cote, E. 2017. A Survey of Learning in Multiagent Environments: Dealing with Non-Stationarity. *arXiv preprint arXiv:1707.09183*.

Jain, M.; Korzhyk, D.; Vaek, O.; Conitzer, V.; Pechouček, M.; and Tambe, M. 2011. A Double Oracle Algorithm for Zero-Sum Security Games on Graphs. In *Autonomous Agents and Multiagent Systems*, 327–334.

Kaelbling, L. P.; Littman, M. L.; and Cassandra, A. R. 1998. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence* 101:99–134.

Katt, S.; Oliehoek, F. A.; and Amato, C. 2017. Learning in POMDPs with Monte Carlo Tree Search. In *International Conference on Machine Learning*.

Klima, R.; Lisy, V.; and Kiekintveld, C. 2015. Combining Online Learning and Equilibrium Computation in Security Games. *International Conference on Decision and Game Theory for Security* 130–149.

Klima, R.; Tuyls, K.; and Oliehoek, F. 2016. Markov Security Games: Learning in Spatial Security Problems. *NIPS Workshop on Learning, Inference and Control of Multi-Agent Systems* 1–8.

Korzhyk, D.; Yin, Z.; Kiekintveld, C.; Conitzer, V.; and Tambe, M. 2011. Stackelberg vs. Nash in Security Games: An Extended Investigation of Interchangeability, Equivalence, and Uniqueness. *Journal of Artificial Intelligence Research* 41:297–327.

Littman, M. L.; Cassandra, A. R.; and Kaelbling, L. P. 1995. Learning Policies for Partially Observable Environments: Scaling Up. In *International Conference on Machine Learning*, 1–59.

Montesh, M. 2013. Rhino Poaching: A New Form of Organised Crime. Technical report, College of Law Research and Innovation Committee of the University of South Africa.

Pita, J.; Jain, M.; Marecki, J.; Ordonez, F.; Portway, C.; Tambe, M.; Western, C.; Paruchuri, P.; and Kraus, S. 2008. Deployed ARMOR Protection: The Application of a Game Theoretic Model for Security at the Los Angeles International Airport. In *International Joint Conference on Autonomous Agents and Multiagent Systems*, volume 3, 1805–1812.

Pita, J.; Jain, M.; Tambe, M.; Ordóñez, F.; and Kraus, S. 2010. Robust Solutions to Stackelberg Games: Addressing Bounded Rationality and Limited Observations in Human Cognition. *Artificial Intelligence* 174(15):1142–1171.

Robinson, J. 1951. An Iterative Method of Solving a Game. *The Annals of Mathematics* 54(2):296–301.

Ross, S.; Chaib-draa, B.; Pineau, J.; Chaib-draa, B.; and Pineau, J. 2007. Bayes-adaptive POMDPs. *Advances in Neural Information Processing Systems* 1225–1232.

Shapley, L. S. 1953. Stochastic Games. *Proceedings of the National Academy of Sciences of the United States of America* 39(10):1095–100.

Shieh, E.; An, B.; Yang, R.; Tambe, M.; Baldwin, C.; DiRenzo, J.; Maule, B.; and Meyer, G. 2012. PROTECT: A Deployed Game Theoretic System to Protect the Ports of the United States. *International Conference on Autonomous Agents and Multiagent Systems* 1:13–20.

Tuyls, K., and Weiss, G. 2012. Multiagent Learning: Basics, Challenges, and Prospects. *AI Magazine* 33(3):41–52.

Wiering, M., and van Otterlo, M. 2013. *Reinforcement Learning: State-of-the-Art*. Springer.

Towards AI that Can Solve Social Dilemmas

Alexander Peysakhovich[*]
Facebook AI Research

Adam Lerer[*]
Facebook AI Research

Abstract

Many scenarios involve a tension between individual interest and the interests of others. Such situations are called social dilemmas. Because of their ubiquity in economic and social interactions constructing agents that can solve social dilemmas is of prime importance to researchers interested in multi-agent systems. We discuss why social dilemmas are particularly difficult, propose a way to measure the 'success' of a strategy, and review recent work on using deep reinforcement learning to construct agents that can do well in both perfect and imperfect information bilateral social dilemmas.

Introduction

How can an agent construct a good strategies for an environment which involves another agent? An early answer to this question was given by (Brown 1951) who considered the idea of 'fictitious play' - an agent is going to play some game once with another agent, if they have access to the game beforehand and they can iterate the game repeatedly in their own mind (ie. during the training phase) and use the strategies they discovered when faced with a real partner (ie. during the test phase). This idea, also called 'self-play', has become an important part of the artificial intelligence toolkit. Self-play where agents try to maximize their own rewards can lead to superhuman performance in zero-sum games like Backgammon (Tesauro 1995), poker (Brown, Ganzfried, and Sandholm 2015), or Go (Silver et al. 2016; 2017) but can lead to bad outcomes in general-sum environments (Sandholm and Crites 1996; Fudenberg and Levine 1998; Lerer and Peysakhovich 2017; Foerster et al. 2017c; Leibo et al. 2017). Recent work has begun to study modified self-play methods to construct good strategies for social dilemmas. In this short note we will review some recent results in this field.

First, we need to determine what it means to do well in a social dilemma. The repeated Prisoner's Dilemma (rPD) is perhaps the most studied social dilemma and gives us a good starting point. In the rPD conditionally cooperative strategies such as Tit-for-Tat (Axelrod 2006) or Win-Stay-Lose-Shift (Nowak and Sigmund 1993) perform well because they reward cooperation today with cooperation tomorrow and so stabilize cooperation while avoiding exploitation. These strategies are studied so heavily because they have intuitively appealing properties. They are nice (begin by cooperating), are simple to explain to a partner, cooperate with cooperators, do not get exploited by defectors, are forgiving (eventually return to cooperation if it breaks down). Importantly, if one can commit to them, they create incentives for a partner to behave cooperatively. A natural desiderata then is to ask for agents in complex social dilemmas that maintain the good properties of these well known PD strategies.

There are several issues in extending these ideas to more complex settings. First, in Markov games 'cooperation' and 'defection' are no longer single acts, but rather sequences of choices (Leibo et al. 2017; Peysakhovich and Lerer 2017a; Lerer and Peysakhovich 2017; Foerster et al. 2017c; Littman 2001). Here agents that want to maintain cooperation within the confines of a single game have to 1) infer whether their partner is cooperating or not, and 2) know how to respond to both of these contingencies. The work we survey here tries to bring ideas from repeated game theory (Fudenberg and Maskin 1986; Dutta 1995; Littman and Stone 2005; De Cote and Littman 2012) to the one-shot setup. There are several issues to overcome: first, rather than maintaining good outcomes by threats of different behavior in *the next iteration* of the game, agents must behave intelligently *within a single game*; second, multiple strategies may be outcome equivalent (e.g. going left then up or up and then left in a grid world); third, function approximation may lead to noise in implementation. We would like to adapt the ideas from repeated game theory to construct strategies that are robust to these issues.

The first set of results we focus on construct conditional cooperators for fully observed games (Lerer and Peysakhovich 2017). The paper in question introduces approximate Markov Tit-for-Tat (amTFT) which applies modified self-play to learn two policies at training time: a fully cooperative policy and a 'safe' policy (we refer to this as defection)[1] which forms an equilibrium with lower payoffs than cooperation.

At test time, the amTFT agent is matched with a partner

[*] Equal contribution. Author order was determined via random.org.

[1] In the PD this action is 'defect'. However, in social dilemmas that occur naturally in economic situations, such a safe policy is the outside option of 'stop transacting with this agent.'

whose policy is unknown. At each time step the amTFT agent computes the gain from the action their partner actually chose compared to the one prescribed by the cooperative policy. This can be done either using a learned Q function or via policy rollouts. We refer to this as a per period debit. If the total debit is below a threshold the agent behaves according to the cooperative policy. If the debit at some time period is above the threshold, the agent switches to the defecting policy for k turns and then returns to cooperation. This k is computed such that the partner's gains (debit) are smaller than the losses they incur (k lost turns of cooperation). The threshold trades off robustness to noise and function approximation with allowing the amTFT agent to be slightly exploitable. It is shown both analytically and experimentally that amTFT can maintain cooperation and avoid being exploited in social dilemmas, including ones where agents learn from raw pixels.

Recent work has argued that TFT-like properties need not be hardwired and strategies can be trained from scratch. Foerster et al. modifies policy gradient to take into account that one's partner is a reactive (rather than static) agent. This method can construct cooperation maintaining strategies in several Markov games. This approach is computationally challenging and has no known theoretical guarantees, and it may construct strategies that are hard to explain (e.g. to a human partner). Despite these drawbacks we believe end-to-end training is a fruitful direction for future research and that explicit constructions like the ones we discuss here are a complement to, not a substitute for, end-to-end approaches.

An advantage of amTFT is that it requires no additional machinery beyond what is required by standard self-play, thus if deep RL can construct competitive agents in an environment such as Atari (Mnih et al. 2015) then we can also construct agents that solve social dilemmas in that environment. A disadvantage is that it requires full observability of a partner's action as well as a good model of the future consequences of a partner's action. Thus, it will not work in many POMDPs. amTFT's focus on future expected rewards as the result of an action can be replaced by consequentialism (Peysakhovich and Lerer 2017a): focusing on the reward stream that one actually obtains. Consequentialist conditionally cooperative (CCC) use self play to compute cooperate and defect strategies like amTFT. CCC uses rollouts of these strategies to compute a time-dependent payoff threshold, if the CCC agent's payoff at a period is below this threshold they defect, otherwise they cooperate. As long as a POMDP satisfies a technical conditions (reward ergodicity) CCC agents can maintain cooperation in the long-run.

CCC is much simpler to compute than amTFT and can perform just as well in some perfect information games. However, this is not always the case. Consider a situation where a partner tries to cheat (very obviously) but due to stochasticity in the environment fails to do so. amTFT would correctly mark this as a deviation from cooperation (because it focuses on the 'intention' behind an action) while CCC would not (because it only looks at consequences). In reality intention is usually somewhat observed (but not perfectly) while consequences are also noisy. This suggests that an important future direction towards constructing agents that solve social dilemmas is finding ways to combine intention and consequences efficiently.

We now describe in more technical detail the results we have surveyed here. Note that the experiments described here are not new, rather they are taken from the papers in question and presented in a summarized way to convey our main points. We point the interested reader back to the original papers for the full details.

Cooperation With Perfect Information

We begin with a generalization of Markov decision problems:

Definition 1 ((Shapley 1953)) *A (finite, 2-player) Markov game consists of*

- *A set of states $S = \{s_1, \dots, s_n\}$*
- *A set of actions for each player $\mathcal{A}^1 = \{a^1_1, \dots, a^1_k\}$, $\mathcal{A}^2 = \{a^2_1, \dots, a^2_k\}$*
- *A transition function $\tau : S \times A_1 \times A_2 \to \Delta(S)$ which tells us the probability distribution on the next state as a function of current state and actions*
- *A reward function for each player $R_i : S \times A^1 \times A^2 \to \mathbb{R}$ which tells us the utility that player gains from a state, action tuple*

We assume rewards are bounded above and below. Players can choose between policies which are maps from states to probability distributions on actions $\pi_i : S \to \Delta(\mathcal{A}_i)$. We denote by Π_i the set of all policies for a player.

Definition 2 *A value function for a player i inputs a state and a pair of policies $V^i(s, \pi^1, \pi^2)$ and gives the expected discounted reward to that player from starting in state s. We assume agents discount the future with rate δ which we subsume into the value function.*

We will be talking about strategic agents so we often refer to the concept of a best response:

Definition 3 *A policy for agent j denoted π_j is a best response starting at state s to a policy π_i if for any π'_j and any s' along the trajectory generated by these policies we have*

$$V^j(s', \pi^i, \pi^j) \geq V^j(s', \pi^i, \pi'^j).$$

We denote the set of such best responses as $BR^j(\pi^i, s)$. If π_j obeys the inequality above for any choice of state s we call it a perfect best response.

The set of stable states in a game is the set of equilibria. We call a policy for player 1 and a policy for player 2 a Nash equilibrium if they are best responses to each other. We call them a Markov perfect equilibrium if they are perfect best responses.

We are interested in a special set of policies:

Definition 4 *Cooperative Markov policies starting from state s (π^1_C, π^2_C) are those which, starting from state s, maximize*

$$V^1(s, \pi^1, \pi^2) + V^2(s, \pi^1, \pi^2).$$

We let the set of cooperative policies be denoted by $\Pi^C_i(c)$. Let the set of policies which are cooperative from any state be the set of perfectly cooperative policies.

A social dilemma is a game where there are no cooperative policies which form equilibria. In other words, if one player commits to play a cooperative policy at every state, there is a way for the other to exploit them and earn higher rewards. Note that in a social dilemma there may be policies which achieve the *payoffs* of cooperative policies because they cooperate on the trajectory of play and prevent exploitation by threatening non-cooperation on states which are never reached by the trajectory.

For the same situation the choice of state representation can affect whether a social dilemma is solvable or unsolvable. To make this more clear, let us consider the repeated Prisoner's Dilemma. In the simplest version rPD individuals are matched to play infinitely many rounds of a stage game in which each player chooses in each round either to give the other player a benefit b at a cost c to themselves (cooperate) or not (defect). When $b > c$ the highest total payoff is achieved when both individuals cooperate, however, each can do better in the short-run by defecting.

The rPD as described in words above can be written as a Markov game in many ways. For example, we can say that there is a single state and two actions per period. In this case, the rPD is an unsolvable social dilemma. This is because the only way to deter defection today is to affect the future payoffs of the defecting agent. With single state, this is impossible. On the other hand, if we model the rPD as a Markov game where the state is the outcome from last period, there are now policies which maintain cooperation and are an equilibrium. For any state representation can never be equilibria which cooperate at *every* state in the rPD because deterring defection today depends on being willing to withhold cooperation from defectors tomorrow and so policies that maintain cooperation at some states must defect at others.

The distinction made above is important because in many examples of interest the simplest choice of representation may not be one that makes the dilemma solvable. In particular, this implies that to play from raw pixels some memory is required, either in the form of an RNN (or similar) or a hardcoded summary statistic. Note that adding memory can create equilibrium policies which maintain cooperation. However, it does not remove equilibria in which both players which always defect. Thus, even with memory applying the self-play paradigm of 'learn a Nash equilibrium at training time and then play your half at test time' may still lead to defecting agents. It has been demonstrated several times that such defecting equilibria can be more robust attractors than cooperative equilibria.

amTFT bypasses this problem by doing the following. When paired with an actual partner the amTFT agent starts in a C phase. While in a C phase the agent behaves according to π^C. However, at each time step while in the C phase the amTFT agent looks at the actions a partner (called j) takes and computes

$$d = Q^j_{CC}(s, \pi^C_i(s), a_j) - Q^j_{CC}(s, \pi^C_i(s), \pi^j_C(s)).$$

If $d > 0$ the amTFT agent switches to a D phase for k periods which is computed such that the loss to the partner from k periods of π^D followed by mutual π^C is relative to both behaving according to π^C the whole time is greater than d. In other words, if a partner deviates today, they lose k periods of cooperation tomorrow.

In Lerer and Peysakhovich, the following analytical result is shown:

Theorem (Intuitive Version) 1 *If the game satisfies some technical conditions which generalize the notion of a Prisoner's Dilemma then if agent j's partner is an amTFT agent, the best response for agent j to play according to π^C_j during the C phase and π^D_j during the D phase. This means that if agents start in a C phase they cooperate forever. If agents start in a D phase they eventually return to cooperation and cooperate forever.*

amTFT is implemented using deep reinforcement learning. Importantly, during training time the amTFT agent has to find cooperative policy π^C and a defect policy π^D. These are found using a modified self-play procedure where the agent either controls both agents and reinforces at each time step on the agents' individual rewards (this is standard self-play and is used to find π^D) or on the joint reward (this finds the joint payoff maximizing policies π^C). In addition, d and k are computed by rollouts and to deal with issues of function approximation d is aggregated over multiple time steps of the game and the D phase begins only if the sum of d passes a threshold.

Cooperation Without Perfect Information

With imperfect information we can use the generalization of a POMDP to the multi-agent case. Here, we take the Markov game definition above and append the notion of observational states. Each player has a set of possible observations O_i and a function Ω_i which maps the state and actions at a given time period to an observation. When Ω_i is the identity for all players we get back a Markov game. Policies, instead of being able to condition on the state, must condition only on observations.

Note that here amTFT is not implementable since the action of a partner may not be perfectly observed. An ideal solution may be to construct a full posterior belief on actions using Bayesian methods. However, often such solutions are intractable. It is possible to construct a simple strategy for any game which satisfies a reward-ergodicity condition: for any pair of policies, there exists a limiting average rate of rewards which is independent of initial starting state. Let ρ_{CC} be the asymptotic rate under joint cooperation and ρ_{CD} be the asymptotic rate under the CCC agent cooperating and the other defecting. We can construct a consequentialist conditionally cooperative (CCC) agent who looks at their current average per period payoff and cooperates if this is above $\alpha\rho_{CC} + (1 - \alpha)\rho_{CD}$ and defects otherwise. This gives a theoretical result:

Theorem (Intuitive Version) 2 *If the game satisfies some technical conditions on the strategies then if a CCC agent is paired with a cooperator they are both guaranteed their cooperative payoffs and if a CCC agent is paired with a defector the defector is guaranteed at most the joint defect payoffs.*

In practice we can construct CCC agents using the same modified self-play as amTFT during training to compute π^C and π^D. Rollouts are used to compute per-period thresholds (Peysakhovich and Lerer 2017a). Note that the analytic results are asymptotic in nature and use the ergodicity condition heavily. To make the CCC strategy work well in finite time we need to use batches of rollouts and suggest using statistics other than the mean (e.g. quantiles) from these batches to construct thresholds. This allows the strategy to trade off flexibly between finite time false positives (assuming a partner is defecting when they are not) and false negatives (missing a defector). Note that this is a purely finite-time tradeoff - asymptotically the theoretical guarantees continue to hold. See the original paper for more details.

Experiments

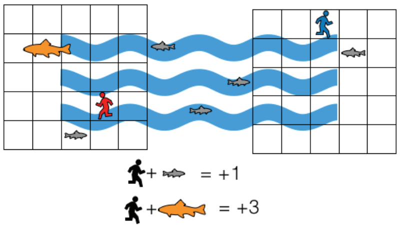
We show the results of applying CCC and amTFT to several games. Here all results are trained using deep RL using standard methods. We refer the readers to the original papers for the full training details.

We also follow the metrics introduced in the original papers. We focus on the key desiderata: a good strategy should be safe from a defector partner, should incentivize cooperation from its partner, and, when matched with a conditional cooperator, should achieve good payoffs.

We define $S_i(X, Y)$ as the expected reward to policy π_1^X matched with π_2^Y. Safety$(X) = S_1(X, D) - S_1(D, D)$ measures how a strategy is safe from exploitation by a defector; and IncentC$(X) = S_2(X, C) - S_2(X, D)$ measures whether a strategy incentivizes cooperation from its partner. While we cannot enumerate all possible conditionally cooperative strategy, we can use a proxy in the case of CCC/amTFT. SelfMatch$(X) = S_1(X, X)$ measures whether a strategy achieves good outcomes with itself. We can compare this payoff to $S_1(C, C)$ and see how much cooperation these policies can achieve.

We begin with the results from using CCC in a POMDP: Fishery. Fishery is a grid-world partially observed Markov game where two agents live on 5×5 grids on opposite sides of a lake. Agents cannot observe the other side of the lake. Fish spawn in each agent's grid and start as young, if they are not caught when they are young they swim to the other side of the lake and become mature. Moving over a fish catches it. Catching a young fish is worth 1 point and catching a mature fish is worth 3 points. Thus, cooperative strategies are those which one catch mature fish but selfish agents are tempted to increase their payoff at a cost to their partner by catching young fish as well. Because this is a partially observed game, we can only use CCC as a cooperation maintaining strategy. We see that in this game agents that play the cooperative strategy (found by modified self-play with both agents receiving the joint reward at training time) can be exploited by defectors (this strategy is found by standard self-play). However CCC achieve cooperation with other CCC agents, is safe, and can incentivize its partner to cooperate.

We now show the results of applying amTFT and CCC to a social dilemma where agents are trained directly from raw pixels. We can change the payoffs of Atari Pong to make it a social dilemma - the Pong Player's Dilemma or PPD (Tampuu et al. 2017). In the PPD when a player scores they receive a reward of 1 while their partner receives a reward of -2. Thus, cooperative strategies are those which gently hit the ball back and forth until the end of the game (and are exploitable by defectors who try hard to score). We see in Figure 2 that both amTFT and CCC perform well in the PPD - cooperating with cooperators, not being exploited by defectors, earning high payoffs when matched with other conditionally cooperative strategies, and incentivizing cooperation from a partner who can choose a strategy.



Strategy	SelfMatch	Safety	IncentC
π^C	141	-36	-31
π^D	64	0	-34
CCC	125	-3	64

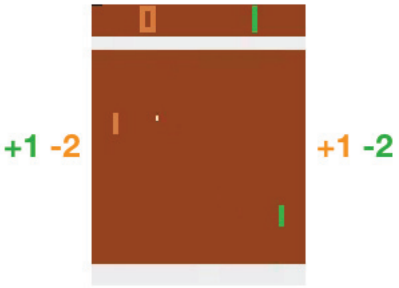
Figure 1: Fishery is a partially observed Markov social dilemma. Mutual cooperation leads to high payoffs but cooperators can be exploited by defectors. CCC cooperates with cooperators, is not exploited by defectors, and makes cooperation a high payoff strategy for its partner.

Because CCC is computationally simpler, one may believe the last result implies it is strictly better than amTFT. This is not always the case. We can change the payoff structure of the PPD to make it stochastic – when a player scores a point their partner gets a reward of $-\frac{2}{p}$ with probability p. We call this the risky PPD. Thus, the expected reward is the same as in the PPD but if p is low then most of the time the cooperative and defect trajectories look identical from the point of view of the payoffs. Here, CCC can be exploited by a defector while amTFT (which uses expected future payoffs) behaves the same as in the standard PPD.

Future Directions

Humans are remarkably adapted to solving bilateral social dilemmas. We have focused on recent work that tries to use deep reinforcement learning to give artificial agents this capability. We have shown that amTFT and CCC can maintain cooperation and avoid exploitation in Markov games. In addition we have discussed the training of these strategies and shown that it requires no more than modified self-play. We now highlight important future directions.

The first is game theoretic. We have discussed a conditionally cooperative strategy that uses the intentions behind an action (amTFT) and one purely uses the consequences (CCC). In the real world intentions are generally only partially observed (either because actions are only partially observed or because modeling their future consequences is difficult)



PPD

Strategy	SelfMatch	Safety	IncentC
π^C	0	-18.4	-12.3
π^D	-5.9	0	-18.4
CCC	0	-4.6	3.3
amTFT	-1.6	-5.2	2.6

Risky PPD

Strategy	SelfMatch	Safety	IncentC
π^C	-0.7	-23.6	-12.8
π^D	-5.8	0	-22.6
CCC	-0.2	-12.2	-5.7
amTFT	-3.6	-3.1	2.5

Figure 2: In the PPD both amTFT and CCC agents can be trained from raw pixels. Cooperators can again be exploited by defectors and conditionally cooperative strategies can be both safe and incentivize cooperation. In the non-stochastic version CCC does as well as amTFT but in the stochastic version CCC can be exploited in finite time games while amTFT cannot.

while consequences can sometimes be poor diagnostics for intentions (because of stochasticity). Thus, an important future direction is to construct strategies that combine these two signals.

The second has to do with non-degeneracy of cooperative strategies. The technical conditions for amTFT and CCC to work require the cooperative strategies satisfy a form of exchangeability - that is, given two sets of cooperative policies any re-combination of them leads to the same outcomes. If cooperative policies are not exchangeable we will have both a social dilemma (‘should we cooperate?’) and a coordination (‘in which way should we cooperate?’) problem. This is strongly related to work on focal points as well as choosing equilibria in coordination games (Schelling 1980; Peysakhovich and Lerer 2017b). Solving this problem, e.g. via introducing communication, is an important avenue for future work. See Kleiman-Weiner et al. for a more in depth discussion.

The third is algorithmic. Any conditionally cooperative strategy needs access to the cooperative strategy and a ‘threat’ strategy. In the surveyed papers we used modified self-play to find these strategies. However, to the best of our knowledge there are no guarantees that even if such strategies exist that standard self-play will find them. In addition, self-play can have stability issues in multi-agent systems as the environment from the perspective of a single agent becomes non-stationary due to the fact that other agents are learning (Foerster et al. 2017b; 2017a; Lowe et al. 2017). Finally, in some situations it can be difficult to find the joint payoff maximizing cooperative policy. Dealing with each of these issues is an important step in scaling these ideas to new environments.

The final issue has to do with human psychology. Here we have focused on implementing strategies that achieve socially optimal payoffs (that is, maximizing the sum of payoffs). However, if we are interested in agents that interact with humans this may not be enough. Human social preferences are more complex than this and the kinds of allocations that humans find fair vary greatly among cultures and contexts – sometimes it is fair for one person to get a lot more than the other and other times it is not (Roth et al. 1991; Henrich et al. 2001; Herrmann, Thöni, and Gächter 2008; List 2007). Perceptions of fairness greatly influence behavior and in particular humans are often willing to pay costs to retaliate against an unfair partner (Camerer and Thaler 1995; Fehr and Gächter 2002; Ouss and Peysakhovich 2015). Thus, if an artificial agent tries to behave according to an efficient but unfair policy, it may find itself stuck in π^D even though a better outcome was possible. Understanding social preferences in context is thus an important question to answer if we seek to construct systems which lead to good outcomes (Crandall et al. 2017; Shirado and Christakis 2017; Hauser et al. 2014).

References

Axelrod, R. M. 2006. *The evolution of cooperation: revised edition*. Basic books.

Brown, N.; Ganzfried, S.; and Sandholm, T. 2015. Hierarchical abstraction, distributed equilibrium computation, and post-processing, with application to a champion no-limit texas hold’em agent. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 7–15. International Foundation for Autonomous Agents and Multiagent Systems.

Brown, G. W. 1951. Iterative solution of games by fictitious play. *Activity analysis of production and allocation* 13(1):374–376.

Camerer, C., and Thaler, R. H. 1995. Anomalies: Ultimatums, dictators and manners. *The Journal of Economic Perspectives* 9(2):209–219.

Crandall, J. W.; Oudah, M.; Ishowo-Oloko, F.; Abdallah, S.; Bonnefon, J.-F.; Cebrian, M.; Shariff, A.; Goodrich, M. A.; Rahwan, I.; et al. 2017. Cooperating with machines. *arXiv preprint arXiv:1703.06207*.

De Cote, E. M., and Littman, M. L. 2012. A polynomial-time nash equilibrium algorithm for repeated stochastic games. *arXiv preprint arXiv:1206.3277*.

Dutta, P. K. 1995. A folk theorem for stochastic games. *Journal of Economic Theory* 66(1):1–32.

Fehr, E., and Gächter, S. 2002. Altruistic punishment in humans. *Nature* 415(6868):137–140.

Foerster, J.; Farquhar, G.; Afouras, T.; Nardelli, N.; and Whiteson, S. 2017a. Counterfactual multi-agent policy gradients. *arXiv preprint arXiv:1705.08926*.

Foerster, J.; Nardelli, N.; Farquhar, G.; Torr, P.; Kohli, P.; Whiteson, S.; et al. 2017b. Stabilising experience replay for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1702.08887*.

Foerster, J. N.; Chen, R. Y.; Al-Shedivat, M.; Whiteson, S.; Abbeel, P.; and Mordatch, I. 2017c. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*.

Fudenberg, D., and Levine, D. K. 1998. *The theory of learning in games*, volume 2. MIT press.

Fudenberg, D., and Maskin, E. 1986. The folk theorem in repeated games with discounting or with incomplete information. *Econometrica: Journal of the Econometric Society* 533–554.

Hauser, O. P.; Rand, D. G.; Peysakhovich, A.; and Nowak, M. A. 2014. Cooperating with the future. *Nature* 511(7508):220–223.

Henrich, J.; Boyd, R.; Bowles, S.; Camerer, C.; Fehr, E.; Gintis, H.; and McElreath, R. 2001. In search of homo economicus: behavioral experiments in 15 small-scale societies. *The American Economic Review* 91(2):73–78.

Herrmann, B.; Thöni, C.; and Gächter, S. 2008. Antisocial punishment across societies. *Science* 319(5868):1362–1367.

Kleiman-Weiner, M.; Ho, M. K.; Austerweil, J. L.; Michael L, L.; and Tenenbaum, J. B. 2016. Coordinate to cooperate or compete: abstract goals and joint intentions in social interaction. In *Proceedings of the 38th Annual Conference of the Cognitive Science Society*.

Leibo, J. Z.; Zambaldi, V.; Lanctot, M.; Marecki, J.; and Graepel, T. 2017. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, 464–473. International Foundation for Autonomous Agents and Multiagent Systems.

Lerer, A., and Peysakhovich, A. 2017. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068*.

List, J. A. 2007. On the interpretation of giving in dictator games. *Journal of Political economy* 115(3):482–493.

Littman, M. L., and Stone, P. 2005. A polynomial-time nash equilibrium algorithm for repeated games. *Decision Support Systems* 39(1):55–66.

Littman, M. L. 2001. Friend-or-foe q-learning in general-sum games. In *ICML*, volume 1, 322–328.

Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; and Mordatch, I. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. *arXiv preprint arXiv:1706.02275*.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518(7540):529–533.

Nowak, M., and Sigmund, K. 1993. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature* 364(6432):56.

Ouss, A., and Peysakhovich, A. 2015. When punishment doesn't pay: 'cold glow' and decisions to punish. *Journal of Law and Economics* 58(3).

Peysakhovich, A., and Lerer, A. 2017a. Consequentialist conditional cooperation in social dilemmas with imperfect information. *arXiv preprint arXiv:1710.06975*.

Peysakhovich, A., and Lerer, A. 2017b. Prosocial learning agents solve generalized stag hunts better than selfish ones. *arXiv preprint arXiv:1709.02865*.

Roth, A. E.; Prasnikar, V.; Okuno-Fujiwara, M.; and Zamir, S. 1991. Bargaining and market behavior in jerusalem, ljubljana, pittsburgh, and tokyo: An experimental study. *The American Economic Review* 1068–1095.

Sandholm, T. W., and Crites, R. H. 1996. Multiagent reinforcement learning in the iterated prisoner's dilemma. *Biosystems* 37(1-2):147–166.

Schelling, T. C. 1980. *The strategy of conflict*. Harvard university press.

Shapley, L. S. 1953. Stochastic games. *Proceedings of the national academy of sciences* 39(10):1095–1100.

Shirado, H., and Christakis, N. A. 2017. Locally noisy autonomous agents improve global human coordination in network experiments. *Nature* 545(7654):370–374.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of go with deep neural networks and tree search. *Nature* 529(7587):484–489.

Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *Nature* 550(7676):354–359.

Tampuu, A.; Matiisen, T.; Kodelja, D.; Kuzovkin, I.; Korjus, K.; Aru, J.; Aru, J.; and Vicente, R. 2017. Multiagent cooperation and competition with deep reinforcement learning. *PloS one* 12(4):e0172395.

Tesauro, G. 1995. Temporal difference learning and td-gammon. *Communications of the ACM* 38(3):58–68.

LfD Training of Heterogeneous Formation Behaviors

William Squires, Sean Luke
Department of Computer Science
George Mason University
4400 University Dr
Fairfax, Virginia 22030

Abstract

Problem domains such as disaster relief, search and rescue, and games can benefit from having a human quickly train coordinated behaviors for a diverse set of agents. Hierarchical Training of Agent Behaviors (HiTAB) is a Learning from Demonstration (LfD) approach that addresses some inherent complexities in multiagent learning, making it possible to train complex heterogeneous behaviors from a small set of training samples. In this paper, we successfully demonstrate LfD training of formation behaviors using a small set of agents that, without retraining, continue to operate correctly when additional agents are available. We selected training of formations for the experiments because formations: require a great deal of coordination between agents, are heterogenous due to the differing roles of participating agents, and can scale as the number of agents grows. We also introduce some extensions to HiTAB that facilitate this type of training.

Introduction

Multiagent Learning from Demonstration (LfD) promises to allow a human to quickly train coordinated behaviors for a diverse set of agents in an online manner with the goal of producing combined behaviors that are more beneficial than agents acting concurrently but without coordination. To do this, Multiagent LfD typically draws on knowledge of each agent's sensors, behaviors, and of the problem domain. This research applies to problem domains in which agents or robots must be rapidly put to use, such as disaster relief, search and rescue, and games where players control a large and diverse set of agents. However, multiagent LfD is very sparsely researched, in large part due to the inherent complexities of multiagent learning due to the Curse of Dimensionality and what we refer to as the *Multiagent Inverse Problem*.

The Curse of Dimensionality states that the number of training samples required for effective machine learning increases exponentially with the dimensionality of the feature vector, which is likely exacerbated by heterogeneity due to increased variety in sensors. The Multiagent Inverse Problem is encountered when trying to learn the appropriate combination of individual agent behaviors to achieve the desired macro-level behavior: while we might have a function available that maps the individual behaviors to the macro-level behavior (namely, a simulator), we do not have the needed function that maps in the other direction. Such inverse problems are normally overcome using offline optimization methods, such as multiagent reinforcement learning or stochastic optimization, but the high cost of generating training samples by a human trainer makes these challenges much more daunting problem for LfD.

The Hierarchical Training of Agent Behaviors (HiTAB) LfD approach (Luke and Ziparo 2010) has addresses these learning challenges. The Curse of Dimensionality is dealt with through iterative behavior decomposition and manual feature selection. The trainer first decomposes the problem into a hierarchy of subproblems, such as breaking "play kiddie soccer" into "play offense" and "play defense", with (for example) "play offense" further broken down to "acquire ball", "manipulate ball", and "kick to goal", and so on. Training is done on the lowest-level behaviors, then the next level, and so on. This allows HiTAB to project the full joint space of features (sensor information) and actions of the top-level behavior into many smaller behaviors, each with its own much smaller subset of sensor features and actions, and consequently fewer training samples.

HiTAB addresses the inverse problem similarly with the introduction of a virtual controller agent hierarchy that allows the trainer to manually decompose the coordination of behaviors among subordinate agent groups. That is, we manually break the swarm into a hierarchy sub-swarms and sub-sub-swarms etc., each headed by a virtual controller agent (a boss). Then we can train small groups to do simple collective behaviors, then assign each a virtual controller (a boss), then train small groups of bosses to do collective behaviors involving directing their subordinates, and so on. Because we are only training small groups of agents at a time, the gulf between individual micro-level behaviors and the desired emergent macro-level phenomenon is mitigated.

Our research goal is to extend HiTAB to train swarm-like heterogeneous behaviors where the resulting behavior can scale to very large numbers of agents. Further, the training should be accomplished without knowing the precise number of agents available for each heterogenous agent types in operation. Training complexity is reduced by including a minimal number of agents to train the coordinated behaviors, with the end result being a minimal controller hierarchy. However, in

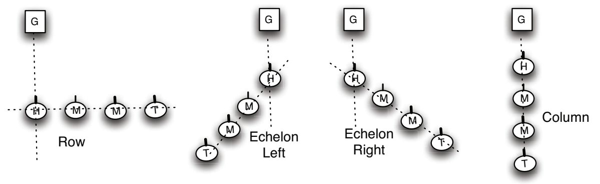


Figure 1: Line Formations

operation the controller hierarchy may need to be grown to effectively utilize large numbers of agents. For example, a grid formation might have a grid controller with some number of subordinate line controllers that are determined by number of available agents.

This paper presents work in progress toward this research goal by training line formations of agents. Line formations are heterogeneous due to differing roles of agents and require considerable coordination to achieve the desired behavior. Formations are also a well studied problem domain in multiagent learning where the effectiveness of the learned behaviors can be visually confirmed. The formations we will learn are shown in Figure 1, each one has a Head agent facing the goal with all other agents forming the line of at some angle to the goal.

Related Work

Heterogeneous Swarms and Hierarchies

Swarm research primarily focuses on the creation of behaviors in which the individual agents only interact with neighboring agents or the environment. Because the interaction is limited in this way, adding agents to the swarm scales with regard to communication. However, this scaling comes at the expense of global information which limits the level of cooperation that can be achieved. Introducing heterogeneity to swarms complicates the problem of having agents perform cooperative behaviors using only local interaction and often introduces a need for structured team organization not present in homogeneous swarms.

In (Elston and Frew 2008) and (Pinciroli et al. 2010), heterogeneous swarms use a hierarchical structure to create coordinated behaviors between an aerial agent and homogeneous sub-swarms. The hierarchy extends the capability of the sub-swarm by leveraging global information relayed to the sub-swarm by the aerial controller. In this case, the sub-swarms still scale because the only additional communication is between the sub-swarm agents and the controller. In (Soule and Heckendorn 2010), an evolutionary approach was presented to learn a controller hierarchy that scales to the available swarm agents with the introduction of force functions that balance agents in the hierarchy or create a new sub-swarms when needed.

Learning from Demonstration

Learning from Demonstration (LfD) is a supervised learning method where training samples are generated through human demonstration (Atkeson and Schaal 1997). LfD literature may be broken into two the categories. In the first category, learning plans or behaviors (Argall et al. 2009), the number of training samples generated by the human demonstrator is generally small since they are only generated when changing behavior or operation. The second category, learning motions or paths (Pastor et al. 2009), often have a large number of samples available as they are generated each time the trajectory changes. HiTAB falls into the first category and so it is focused on effectively learning behaviors from a small number of training samples.

Multiagent LfD produces additional very difficult challenges as previously discussed, and as a result is only lightly researched. In (Chernova and Veloso 2010), robots were trained to cooperatively sort colored balls into the appropriate bin where the robots requested additional demonstration when uncertain of correct action. Additional LfD multi-agent learning involves learning from the joint demonstration of multiple trainers. For example, in (Martins and Demiris 2010) an approach was developed where the individual sequence of actions for each robot are captured and then the sequence of group behaviors is determined through analysis of the individual action sequences over space and time. In (Blokzijl-Zanker and Demiris 2012), robots learn to collaboratively open a door by extracting a template for the behavior and adapting it to doors in other settings. These methods work well for small teams, but become dramatically more complex as more robots are added.

Learning Formation Behaviors

Formations are a well studied problem in learning coordinated multi-agent behaviors, many of them using motor schema (Balch and Arkin 1998) or other potential field based approaches. In (Das et al. 2002), formation control leverages multiple controllers based on vision sensors on all agents. More recent literature has focused on new potential field methods of formation control for swarms (Barnes, Fields, and Valavanis 2009). The formations is this paper, trained as leader-referenced formations, are not intended as a better method of formation control, but as an interesting test problem for heterogeneous multiagent LfD that can be extend to swarm-like behaviors.

Background on HiTAB

HiTAB was introduced in (Luke and Ziparo 2010) originally as a single-agent LfD method for training individual agent behaviors that addresses domain space complexity. HiTAB learns behaviors in the form of hierarchical finite-state automata (HFA), where the states are either atomic agent behaviors or lower level HFA learned earlier. The HFA are defined through manual decomposition of the desired top-level behavior, and the lowest level in the HFA only have atomic agent behaviors as states. For each HFA, the trainer manually selects the required states (subbehaviors) and features, or agent sensors, needed to determine the transition between states. Because the states and features are manually selected by the trainer, HiTAB is only learning the transition function for the HFA. By default the learning is in the form of a C4.5 decision tree.

Behaviors and features may be parameterized and be bound to a *target*, which is some object in the environment. An example for a feature is *DistanceTo(A)*, which is bound the target *ClosestAgent* by the trainer to get the *distance to the closest agent*. An example for a behavior is a trained behavior *Goto(A)*, which is bound to the target *Home Base* resulting in a *go to home base* behavior. Parameters may also be bound to a variable of the HFA so that the learned HFA is parameterized, such as *SpreadBetween(A,B)* defined later in this paper.

HiTAB also has a special atomic behavior called *Done*. Done sets a Done flag, which is accessible through the *Done* feature, and immediately transitions to Start and the flag remains set unless it is specifically clear or the top-level agent behavior is changed. This is useful when some behavior has to be completed before another begins. Done is also special in that it is also an atomic behavior for training controller agents.

Individual Agent Training

When decomposition of the HFA is complete, features have been selected, and parameters bound, the demonstrator can then begin training. While in *training mode* the trainer teleoperates the agent, changing behaviors at the appropriate time. Whenever a behavior is changed, a training sample is created containing the current behavior, the current feature values when the behavior was changed, and the new behavior. Behaviors that are meant to continue until the next behavior change by the trainer will add an additional *continuation* sample with the new behavior, the feature values, and the new behavior again.

The trainer switches to *testing mode* when all training samples have been created, causing the transition function of the HFA to be learned. The trainer then observer the trained behavior in operation and saves it to the behavior library if it is working correctly. Otherwise, the trainer switches back to training mode and provides additional samples and repeats the test mode step.

Homogeneous Multiagent Training

In (Sullivan and Luke 2012), HiTAB was extended to homogeneous multiagent training with the introduction of *virtual controller* agents responsible for coordinating a subordinate group of agents. HiTAB again models the controller's behavior as an HFA, which is decomposed so that at the lowest level the HFA only contain only the atomic behaviors of the controller.

These atomic behaviors correspond to the top-level behaviors of each of the agents in the controller's subordinate group. As such, atomic controller behaviors manipulate the subordinate agents rather than the controller itself, with a transition in the controller HFA directing all subordinates to change their behavior. Controller features are programmed by the trainer to provide statistical information from the features or states of the subordinate agents. As with individual agents, behaviors and features may be parameterized.

To increase cooperation among the homogeneous individual agents, a hierarchy of controllers may be trained. For a higher level controller the training method is the same, with its the atomic behaviors being the trained behaviors of the virtual controller agents in its subordinate group.

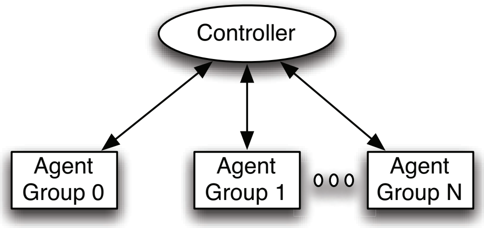


Figure 2: Heterogeneous Controller

Heterogeneous Multiagent Training

In (Sullivan et al. 2015), HiTAB was further extended to heterogeneous multiagent training, where virtual controllers may have more than one subordinate agent group as shown in Figure 2. The subordinate groups of a heterogeneous controller each contain a different agent type. Again the coordinating behavior is represented as an HFA, which is decomposed so that the lowest level HFA has only the atomic behaviors of the controller.

Each atomic behavior for heterogeneous controllers is now a *joint behavior*, which is some permutation of trained behaviors of the subordinate agent groups. A *continuation* behavior may be defined for one or more of the agent groups in the joint behavior, meaning that all agents in that group should continue the previously directed behavior. As in the homogeneous case, all agents within a single subordinate group are running the same behavior. Depending on the number of agent groups and the number of trained behaviors within subordinate agents, the number of joint behavior permutations can be quite lengthy. Additionally, a given permutation of behaviors may not be meaningful to the trainer in the context of the training problem. For this reason it is left to the trainer to define the joint behaviors for the controller, and consequently presented with a meaningful and minimal set of atomic behaviors during training.

Joint behaviors take the form *Name(Behavior 0, Behavior 1, ..., Behavior N)*, where *Behavior i* is a trained behavior of subordinate agent group *i* . For example, Init(Face(X), Surround(X), MoveBetween(X, Y)) is a joint behavior named Init that tells agents in group 0 to Face some target X, agents in group 1 to Surround some target X, and agents in group 2 to MoveBetween two targets X and Y. If a continuation is specified for an agent group, the subordinate behavior is left blank (□), for example SpreadAgents(□, AdjustSeparation(X, Y), SpreadBetween(X, Y)) has no behavior defined for agents in group 0.

While the work in (Sullivan et al. 2015) provided a successful demonstration of training heterogeneous controller agents it falls short of the research under way in this work. The controller hierarchy, shown in Figure 3, is two levels deep where each agent group has a single agent. While this is heterogeneous and mutliagent by definition, it did not demon-

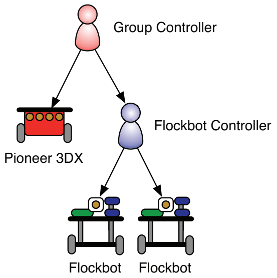


Figure 3: Box-pushing Controller

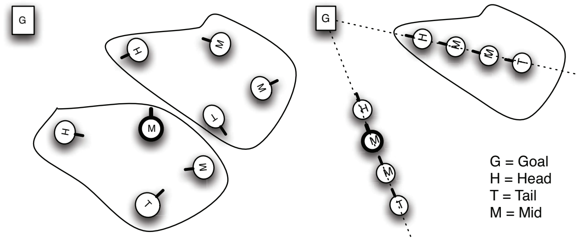


Figure 4: Column Formation

strate training of heterogenous behaviors that scale to some unknown number of agents in operation. Also, because each group contained a single agent there was no need for feature aggregation, which is necessary in many heterogeneous LfD training problems.

Our Extensions to HiTAB

Group Features

Introduced in this work, a *group feature* applies an *aggregator function* to a feature value for all agents within an agent group, or from agents in all subordinate groups if a group isn't specified. An aggregator function is a simple statistical function such as Max, Min, Average, and Range. Group features have the form *Name(group, aggregator, feature)*. For example, MostDistant(0, Max, DistanceTo(X)) defines a group feature named MostDistant whose value is the maximum value of the DistanceTo(X) feature for agents in group 0. For basic agent groups, a group feature can be defined for any feature of the basic agent or a feature common to all subordinate groups when a group is not specified. For controller agent groups, a group feature can be defined for any group feature of the subordinate controller. Because feature values are passed up the hierarchy, this means that group features may need to be defined in a lower level of the hierarchy even though they aren't needed for training coordinated behaviors at that level.

Targets with Context

Targets referencing other agents in the heterogeneous setting may require additional context that wasn't necessary in homogeneous training. For example, the SpreadBetween behavior, which moves an agent between one or more targets X and Y, may require targets X and Y to actually be bound to agents in groups 0 and 1 respectively. For the purposes of this work, the following targets have been defined:

- **Closest Agent - My Group** is the closest agent within the agent's own group.
- **Closest Agent - Group *n*** is the closest agent in subgroup *n* of the agent's controller.
- **Closest Agent - My Parent** is the closest agent in all subgroups of the agent's controller.
- **Second Closest Agent - My Parent** is the second closest agent in all subgroups of the agent's controller.

Joint Behaviors with Parameter Conflicts

Joint behaviors specify one behavior per subordinate group; these behaviors may also be parameterized. When multiple behaviors in a joint behavior are parameterized it is possible that parameters with the same name are not to be bound to the same target. In such cases the trainer can select one of the conflicting behaviors and trivially train a new behavior binding the parameters in the heterogeneous setting. Such conflicts are described in the *InitHT*, *MoveMids*, and *SpreadMids* joint behaviors in the ***Heterogeneous Controller Behavior Training*** section.

For example, in MoveMids the *MoveAwaySome(X,Y)* behavior has parameters that conflict with *MoveBetweenAvoid(X,Y)*, so it is trained for a Mid agent in the heterogenous setting by binding parameter A to *Closest Agent - My Group* and parameter B to *Closest Agent - Group 0*. The trainer then: starts training, selects the MoveAwaySome behavior, ends training, saves the behavior as *CreateSpace*, and replaces *MoveAwaySome(X,Y)* in the joint behavior with *CreateSpace*.

Training Column Formation Behaviors

In this work, training has focused on creating four line formations: column, row, echelon right, and echelon left with respect to some *Goal* in the environment. The controller agent for these problems has three subordinate agent groups to train according to the different agent roles in the formation: Head, Tail, and Mid agents whose agent groups are indexed 0–2 respectively. Individual agents are initially distributed randomly in the environment as shown on the left in Figure 4, and each line formation behavior coordinates the agents through a series of steps to position and orient them with respect to the Goal. The general steps to creating a line formation are shown below.

1. **Initialize** by orienting the Head agent at the Goal and positioning the Tail agent such that the angle between the Goal and Tail from the perspective of the Head agent is some value, depending on the formation.
2. **Move Mid Agents** between the Tail and Head agents
3. **Spread Mid Agents** between Tail and Head and adjust the distance from the Tail to the Head to achieve the desired agent spacing.
4. **Orient Mid and Tail Agents** like the Head agent.

Using this approach, the only difference in training between the four line formations is the **Initialize** step. The desired result of the column behavior is show on the right of Figure 4.

Individual Behavior Training

While the training of the individual behaviors is not the focus of this paper, there are some important points to make about training individual behaviors in light of the overall training problem. First, training complexity in individual behaviors is much preferred over training complexity in controller behaviors because we wish to maximize agent autonomy and consequently minimize communication with the controller. Second, training individual behaviors should should be parameterized to promote reuse. For example, the *MoveBetween* behavior for Mid agents is trained to move the agent between points A and B. The parameters A and B are bound later to *targets* in the heterogeneous setting as described in the next section. In total, we trained 26 individual behaviors to support the line formation behaviors. The behaviors listed below are the individual behaviors used in the definition of joint behaviors for the controller.

- **AlignFront**(X): Orient the agent so that it is facing the target X.
- **GotoAvoid**(X): Move the agent toward the target X while avoiding objects or agents in its path.
- **CircleAvoid**(X): Move the agent in a circle around the target X, with avoidance, in a clockwise direction. The radius of the circle is set as the distance to X at the time the behavior starts.
- **CircleNegAvoid**(X): Move the agent in a circle around the target X, with avoidance, in a counter-clockwise direction. The radius of the circle is set as the distance to X at the time the behavior starts.
- **MoveAwaySome**(X, Y): When the distance to the target X is below a threshold, move the agent a short distance from the opposite direction of target Y, with avoidance.
- **MoveCloserTo**(X, Y): Move the agent in the direction of target Y when the distance to target X is greater than some threshold.
- **MoveBetweenAvoid**(X, Y): Move the agent onto the line between two targets X and Y, with avoidance.
- **SpreadBetween**(X, Y): Move the agent equidistant between any agent in its group or one of the endpoints X and Y if the agent has no other agent between itself and X or Y.
- **AlignLike**(X): Orient the agent in the same direction as the target X.

The state machines used in training two of the more complex individual behaviors are shown in Figure 5 with the associated features defined below the corresponding state HFA. The LineAssoc behavior is intended to run after an agent has moved onto the line between two points (X, Y) and it is assumed that there is at least one other agent on the line and in the same agent group as the training agent. This behavior ensures that the agent is associated with an endpoint if it is directly adjacent, meaning there is no other agent between the agent and the endpoint, and otherwise associated with the closest agent in its agent group. Once the association is made the agent will remain in the Done state until the behavior is changed. The DirectionSimilarity(A,B) feature returns the cosine similarity between the vectors from the agent to A and the agent to B. With all agents being on the line, DS returns a value close to 1 when the closest agent and X are in the same direction and close 0 otherwise. Training LineAssoc was completed using 29 training samples.

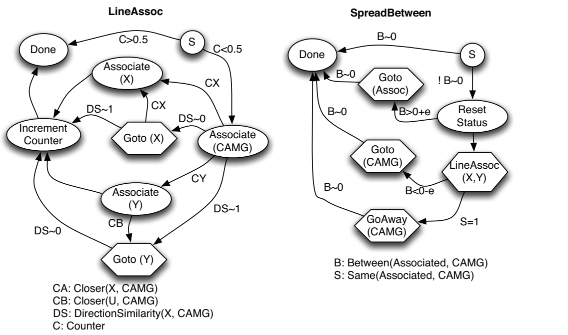


Figure 5: The *LineAssociate* and *SpreadBetween* HFA

The SpreadBetween behavior repositions an agent so that it is equidistant between the associated object and the closest agent in the same agent group. Because it is utilizing the LineAssoc behavior, the associated object is either one of the endpoints or another agent on the line between them. The Between(X, Y) feature returns a value between -1 and 1 with the zero value occurring at the point equidistant from X and Y. If X and Y are the same object, then zero is returned. Training SpreadBetween was completed using 15 training samples.

It is important to note that the Done behavior sets the Done flag and immediately transitions to Start, which is why additional transitions from Start to Done were trained. Also, since a behavior can only be bound to one target a behavior sometimes has to be trivially trained (with no transitions) and saved under a different name. Thus the Goto/Goto2 and Associate/Assoc2/Assoc3 behaviors.

Heterogeneous Controller Behavior Training

Heterogeneous controller agent behaviors use the same behavior decomposition, feature selection, and training method as individual behaviors and homogeneous controllers. However, there are a few additional steps required by the trainer for heterogeneous controllers.

1. Define joint behaviors from the trained individual behaviors of subordinate agents.
2. Perform trivial training of individual behaviors to eliminate parameter conflicts and update joint behaviors to reference the new behaviors.
3. Define group features
4. Bind joint behaviors to targets

5. Bind group features to targets
6. Train controller behavior FSA based on joint behaviors and group features.

Before describing the joint behaviors and group features, it is helpful to define a shorter and more descriptive notation for the common targets for group features and joint behaviors.

- Goal (A): The goal is bound to the parameter A.
- Head (H): The Head agent is *Closest Agent - Group 0*.
- Tail (T): The Tail agent is *Closest Agent - Group 1*.
- Closest Mid (CM): The closest Mid agent is *Closest Agent - Group 2*.
- Closest in Formation (CF): The closest agent in the formation is *Closest Agent - My Parent*.
- Second Closest in Formation (SCF): The second closest agent in the formation is *Second Closest Agent - My Parent*.

Joint Behaviors Seven joint behaviors are defined as atomic behaviors for the training of line formations.

- **InitHT**(AlignFront(X), GotoAvoid(X), □): To eliminate a parameter conflict, Goto(X) is trained as Goto(H) for the Tail agent and saved as GotoHead. The updated joint behavior is InitHT(AlignFront(X), GotoHead, □).
- **AlignTail**(□, CircleAvoid(X), □).
- **AlignTailNeg**(□, CircleNegAvoid(X), □).
- **TailDone**(□, Done, □): Note that Done is a special non-trained behavior that can referenced in a joint behavior.
- **MoveMids**(□, MoveAwaySome(X, Y), MoveBetweenAvoid(X, Y)): To eliminate a parameter conflict, MoveAwaySome(X, Y) is trained as MoveAwaySome(CM, H) for the Tail agent and saved as CreateSpace. The updated joint behavior is MoveMids(□, CreateSpace, MoveBetweenAvoid(X, Y)).
- **SpreadMids**(□, MoveCloserTo(X, Y), SpreadBetween(X, Y)): To eliminate a parameter conflict, MoveCloserTo(X, Y) is trained as MoveCloserTo(CM,H) for the Tail agent and saved as AdjustSeparation. The updated joint behavior is SpreadMids(□, AdjustSeparation, SpreadBetween(X, Y))
- **AlignLikeHead**(□, AlignLike(X), AlignLike(X)): There is no parameter conflict in this case.

Group Features Five group features are defined for the training and operation of the heterogeneous controller FSAs.

- **TailAligned**(0, Average, RelativeDirection(X, Y,Z)): Returns the angle between vectors XY and XZ as measured by the Head agent.
- **MidsBetween**(2, Max, DistanceBetween(X, Y)): Returns the maximum distance from Mid agents to nearest point on the line between X and Y.
- **MidsDistributed**(2, Max, DistanceTo(X)): DistanceTo returns the distance from the Mid agents to X.
- **TailProximal**(1, Max, DistanceTo(X): Returns the distance from the Tail agent to X.
- **MidsDone**(□, Min, Done): Returns 1 if the Done flag is set for all agents in the formation.

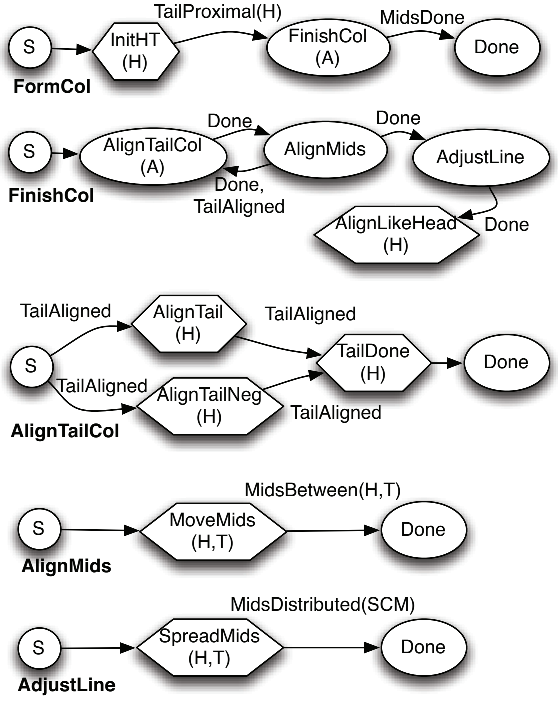


Figure 6: Column Formation HFA

Training and Results

Training was successfully performed for each of the four line formation behaviors. Training of the column formation behavior, FormCol, was completed by training the HFA shown in Figure 6 from the bottom up as pictured. Joint behaviors are represented by a hexagonal shape while trained behaviors (and Done) are elliptical shapes. The targets of the behaviors and features are noted in parentheses. The results of a run of the behavior in the HiTAB simulator with 3 Mid agents are shown in Figure 7 where the Home Base marker is selected as the goal. Training of the entire controller HFA was accomplished with a total of 36 training examples, the bulk of which were used to train AlignTailCol (18) and FinishCol (12).

Training the other formations is very similar by following these steps steps below. The column formation only differs in that the last two steps are not required.

1. Position Head near target and orient so that angle with respect to the goal is that of the desired line formation.
2. Form the other agents in a line behind the head.
3. Orient the Head to face the goal.
4. Orient the other agents like the head.

Because there was some reuse of trained behaviors in column training the additional line formations only required 33 training samples each even with the two additional steps. The goal of this work wasn't to minimize the number of training samples, but the totals indicate the efficacy of the manual

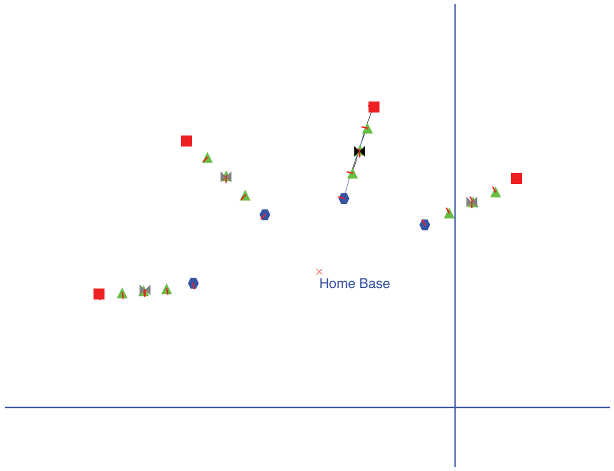


Figure 7: Column Formation in HiTAB

behavior decomposition to address the inverse problem for heterogeneous controller training.

After we trained the behaviors, we ran the learned behavior with different numbers of middle agents. As expected, the formation behavior with additional agents formed a longer line. The Line formation behaviors have the limitation that the controller hierarchy is fixed in size. So while this work accomplished the goal of training heterogeneous line formations that scale as the number of Mid agents grows, it does not (and cannot) grow the number of virtual controllers to effectively use additional Head or Tail agents.

Training the behaviors is complicated a problem related to behavior decomposition and feature selection. When we decompose the behaviors, features are selected based on the state transitions in the HFA and the way those features are used in the learned transition function sometimes differ from the expectation of the trainer because of the underlying machine learning method. As decision trees are the default method to learn the transition function, the small number of training samples often introduces some randomness in choosing which feature is determined to have the most information gain. This becomes more problematic when floating point features are used since they will often have a different value for all training samples. This can be overcome by decomposing behaviors so that they have at most one floating point feature. Using the *Done* behavior allows a higher level HFA to use the *Done* feature in training rather than a floating point feature. The decomposed HFA in Figure 6 and Figure 5 reflect this approach.

Conclusions and Future Work

This work demonstrates the training of complex heterogeneous multiagent behaviors using HiTAB. Specifically, we trained heterogeneous virtual controllers which coordinated subordinate agents to produce four different line formations. Without retraining, the behaviors continue to operate correctly when the number of agents is increased. This training required substantial effort on the part of the trainer to manually decompose the controller behavior, define joint behaviors and group features, and finally to train the controller behavior. However, the training required a very small number of samples and no special purpose code.

For future work, we will focus on training heterogeneous behaviors where controller agent groups may grow in size based on the number of agents available in operation. This is a complex problem since the hierarchy may be deep and unbalanced with a decision to be made at each level to expand the controller agent group. The basic agents then have to be effectively distributed in the expanded controller hierarchy.

Two types of test problem have been identified to further this research, N-deep formations and heterogeneous game scenarios. First, *N-deep formations* require a greater degree of coordination and will most likely present new challenges in terms of contextual agent targets and group features for controllers at level 2 and above in a controller hierarchy. As previously described, a grid formation my require growth of a controller agent group to expand the grid to effectively include the available agents. This problem can be extended to a line of grids, a wedge of grids, and so on.

Second, we will concentrate on *heterogeneous game scenarios*. While formations are an easily understood set of challenge problems for heterogeneous behavior training and have behaviors that can scale to the agents available, it is difficult to measure the effectiveness of growing the controller hierarchy. We will create a game scenario where resulting heterogeneous controller hierarchy can be grown to utilize additional agents and there is also some goal that can be measured. This will allow us to compare the effectiveness of our hierarchy growing algorithm to other methods.

Acknowledgments

This research in this paper was conducted with the support of research infrastructure developed under NSF grant 1727303.

References

Argall, B. D.; Chernova, S.; Veloso, M.; and Browning, B. 2009. A survey of robot learning from demonstration. *Robotics and Autonomous Systems* 57(5):469–483.

Atkeson, C. G., and Schaal, S. 1997. Robot learning from demonstration. In *ICML*, volume 97, 12–20.

Balch, T., and Arkin, R. C. 1998. Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation* 14(6):926–939.

Barnes, L. E.; Fields, M. A.; and Valavanis, K. P. 2009. Swarm formation control utilizing elliptical surfaces and limiting functions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39(6):1434–1445.

Blokzijl-Zanker, M., and Demiris, Y. 2012. Multi robot learning by demonstration. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, 1207–1208. International Foundation for Autonomous Agents and Multiagent Systems.

Chernova, S., and Veloso, M. 2010. Confidence-based multi-robot learning from demonstration. *International Journal of Social Robotics* 2(2):195–215.

Das, A. K.; Fierro, R.; Kumar, V.; Ostrowski, J. P.; Spletzer, J.; and Taylor, C. J. 2002. A vision-based formation control framework. *IEEE Transactions on Robotics and Automation* 18(5):813–825.

Elston, J., and Frew, E. W. 2008. Hierarchical distributed control for search and tracking by heterogeneous aerial robot networks. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, 170–175. IEEE.

Luke, S., and Ziparo, V. A. 2010. Learn to behave! rapid training of behavior automata. In *Proceedings of Adaptive and Learning Agents Workshop at AAMAS 2010*.

Martins, M. F., and Demiris, Y. 2010. Learning multi-robot joint action plans from simultaneous task execution demonstrations. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, 931–938. International Foundation for Autonomous Agents and Multiagent Systems.

Pastor, P.; Hoffmann, H.; Asfour, T.; and Schaal, S. 2009. Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 763–768. IEEE.

Pinciroli, C.; O'Grady, R.; Christensen, A. L.; and Dorigo, M. 2010. Coordinating heterogeneous swarms through minimal communication among homogeneous sub-swarms. In *International Conference on Swarm Intelligence*, 558–559. Springer Berlin Heidelberg.

Soule, T., and Heckendorn, R. B. 2010. A developmental approach to evolving scalable hierarchies for multi-agent swarms. In *Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation*, 1769–1776. ACM.

Sullivan, K., and Luke, S. 2012. Learning from demonstration with swarm hierarchies. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 197–204. International Foundation for Autonomous Agents and Multiagent Systems.

Sullivan, K.; Wei, E.; Squires, B.; Wicke, D.; and Luke, S. 2015. Training heterogeneous teams of robots. In *Autonomous Robots and Multirobot Systems (ARMS)*.

AAAI Press

2275 East Bayshore Road, Suite 160
Palo Alto, CA 94303

9 781577 357957