The Thirty-Second International Florida
Artificial Intelligence Research Society Conference (FLAIRS-32)

A Contextual-Based Framework for Opinion Formation

Eugene Santos, Clement Nyanhongo
Thayer School of Engineering
Dartmouth College
Hanover NH, 03755

Abstract

During opinion formation, interacting agents can be assumed to be engaging in learning and decision-making processes to satisfy their individual goals. These goals are determined by the agents' preferences - which are often unknown, complex and unpredictable. Most opinion formation frameworks however, assume static preferences and fail to model practical situations where human preferences change. We propose a new framework to simulate the process of opinion formation under uncertainty and dynamism. Agents who are unaware of their implicit contextual preferences utilize inverse reinforcement learning to compute reward functions that determines their preferences. Reinforcement learning is subsequently used to optimize the agents' behavior and satisfy their individual goals. The novelty of our approach lies in its ability to capture uncertainty and dynamism in the agent's preferences, which are assumed to be unknown initially. This framework is compared to a baseline method based on reinforcement learning, and results show its ability to perform better under dynamic scenarios.

Introduction

An opinion is defined as a 'personal belief' or 'an unsupported claim' (Damer 2008) about an issue or an object. It cannot be proven and it is based on intrinsic humanistic attributes such as perceptions, emotions, social influence and other cognitive processes (Kuhne 2014). Opinions are dynamic and they change over time as the individual is exposed to new experiences which modify his or her perspective (Lenz 2009). The process of opinion formation can be modeled as a learning and decision-making process in which agents interact to make cognitive actions that satisfy their goals (Yu 2013). These goals are determined by preferences such as the agent's emotions, knowledge base, or perception. Through cognitive actions, agents incorporate or eliminate information to infer opinions about a subject. An example could be an interview scene in which an employer simultaneously interviews a group of students to pick the best candidate. Initially, the employer knows nothing about the students but learns their abilities through opinions that he forms as the interview progresses.

Opinion formation is often studied using consensus rates of interacting individuals in a network. The DeGroot model is a widely known approach that describes the process of reaching consensus by fitting probability distributions of all individuals in the network (1974). A model by Friedkin and Johnsen works by computing weighted averages of social influence that flows within the network (1990). Evolutionary game theory approaches simulate strategic situations for agents who make decisions based on the payoffs that they can potentially get (Ding et al. 2009). Bayesian approaches have also been applied specifically in situations that require knowledge representation to capture the causal relationships between variables (Gu, Santos, and Santos 2013).

These approaches help us to understand opinion dynamics as a consequence of information spread, consensus rate and influence within social groups; however, they overlook conditions of uncertainty and dynamism that exist in opinion formation. Game theory models for example, often perform well in static environments where conditions are well-defined, but poorly in environments where an agent's behavior is dynamic. This often leads to improper training (Yu 2013), a situation where the expected experience differs from the observed experience during opinion formation.

In this paper, we strive to overcome these limitations through the following contributions: 1. We present a framework that aims to find the context which incorporates agent preferences that affect opinion formation. Initially, agents are assumed to be unaware or uncertain of each other's goals, but they acquire the full picture through reward functions that they compute from past interactions. 2. This framework addresses the issue of dynamism and improper training through reward function updates over time. To demonstrate the effectiveness of our model, we compare its performance to a baseline approach discussed in the related works section.

Related Works

In the development of our framework, we adopted an approach described by Yu and Santos, to model opinion formation as both a learning and decision-making process (2016). Agents are defined by goal profiles which quantify their preferences. Agents act to maximize their rewards as determined by these goal profiles. They take actions to modify their knowledge and infer results on a particular subject that they are forming an opinion on. They make optimal actions through policies that they learn by reinforcement learning.

We adopted several parts of this framework due to its rigorous definition of opinion formation as a decision-making process. However, limitations arise from the crafting of a static reward function to determine agent behavior. A typical problem is the issue of improper training that arise when the agent's learning experience differs from the actual testing experience. The agent does not perform well since it learned a reward function that was not designed for the problem that it faces. Our work builds upon this framework to address these challenges.

Technical Background

This section explains the theory behind the main computational tools used in our framework

Reinforcement Learning

Reinforcement learning (RL) enables an agent to learn an optimal policy in an interactive environment (Sutton and Barto 1998). The agent learns through feedback which is provided by the environment in form of rewards or regret. The problem of RL is to find an optimal policy that maximizes the cumulative reward for the agent. This problem is solved by a Markov Decision Process (MDP) defined as a 5-tuple (S, A, R, P, γ) where S represents the states in the environment, A represents the actions that an agent can take, R represents the reward function, P represents the probability transition function between states and γ represents the discount factor. Several algorithms are used to solve RL problems and the most widely used is Q-learning, which works by estimating values of state-action pairs (Watkins and Dayan 1992). Q-learning generates a q-vector with values for each state updated by Equation 1.

$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha\,[R + \gamma * \max Q(s_{t+1}, a)] \quad (1)$$

$Q(s_t, a_t)$ is the value of state s under an action a at time t, $Q(s_{t+1}, a)$ is the value of a potential future state under a future action a, α is the learning rate and γ is the discount factor.

Inverse Reinforcement Learning

Inverse Reinforcement Learning (IRL) was originally proposed by Russell (1998) as a means to derive an agent's reward function from its past behavior. The reward function shows the preferences that an agent utilizes in its demonstrated behavior. IRL takes an MDP without a reward function (MDP/R) and approximates the function. Several methods are utilized to solve IRL problems and they differ based on the mechanisms that they use for learning. In this paper, we analyze the Maximum Entropy (Maxent) IRL algorithm (Ziebart et al. 2008) and the c-neighbor IRL algorithm (Santos et al. 2018). Both algorithms take as input, a set of the agent's past observed trajectories:

$$\{\tau_1, \tau_2, \ldots, \tau_n\}. \quad (2)$$

Each trajectory is defined as a sequence of states and actions such that:

$$\tau_i = \{s_1, a_1, s_2, a_2 \ldots, s_T\}. \quad (3)$$

For Maxent IRL, rewards, R for a particular trajectory are defined as a linear combination of feature values f_τ such that:

$$R(f_\tau) = \theta^T f_\tau \quad (4)$$

θ refers to feature weights which the algorithm aims to find. Maxent IRL uses a probabilistic model to find a solution that maximizes entropy over all possible reward distributions. We decided to use this algorithm since it finds a solution that theoretically guarantees accuracy (Arora and Doshi 2018).

The c-neighbor IRL algorithm searches an agent's trajectory space to find alternative trajectories that the agent did not take, but were close enough to some trajectory from the observed set in Equation 2. Given trajectories:

$$\tau = \{s_i, a_i, s_{i+1}, a_{i+1}, \ldots, s_{i+k}\} \quad (5)$$

$$\tau' = \{s'_i, a'_i, s'_{i+1}, a'_{i+1}, s'_{i+2}, \ldots, s'_{i+k}\} \quad (6)$$

τ' is an m-sized c-neighbor of τ if:

$$\left|\{s'_j | s'_j \neq s_j\} \cup \{a'_j | a'_j \neq a_j\}\right| = m;\ \forall_j \in \{1, \ldots, k\} \quad (7)$$

Using the set of all observed trajectories and the derived c-neighbors, this algorithm uses linear programming methods to find a solution under an optimization constraint which assumes that the observed trajectories have higher value than the unseen neighborhood trajectories that the agent could have taken. We chose this algorithm since it does not treat the reward function as a linear combination of features.

Double Transition Model

A Double Transition Model is a cognitive structure that graphically captures the human decision-making process from past behavior. The DTM was originally proposed by Yu (2013) as a way to describe human opinion formation

from computational simulation and it has been applied to study decision-making styles (Santos et al. 2017). DTM nodes represent cognitive states and edges represents transitions during decision-making. The DTM can be viewed as a cross product of a query transition graph (QTG), and a memory transition graph (MTG). Nodes in the MTG represents an agent's working memory, which comprises of the agent's history of past experiences. The working memory is sequentially updated through learning episodes that the agent gathers as it perceives new information. Nodes in the QTG represents a query at a particular time instant. Each query can be represented as a vector such as $q = [?, ?, 1, 0, ?, ?]$, where each entry represents a value of a feature with '?' denoting unknown attributes that the agent tries to infer a value through a new opinion.

An agent makes cognitive actions to transition between cognitive states hence modifying its working memory. In our opinion formation scenario, cognitive actions can add, maintain or remove learning episodes from the DTM's cognitive states. Since the DTM is sequentially updated during interactions, it provides a means of generating an MDP where cognitive states and transitions have a one-to-one mapping with an equivalent MDP. The DTM constantly updates itself as new information comes, thus giving a dynamic MDP.

Approach

To develop the new framework, we adopted the testbed and technical setup from Yu and Santos' framework (2016). Their framework is the baseline method that we use to evaluate our proposed framework. We use this baseline since it models opinion formation as both a learning and decision-making process. Our proposed framework differs from the baseline by incorporating dynamism through inverse reinforcement learning and reward updating. Other elements such as inference and knowledge representation are implemented differently.

In this section, we first describe the experimental setup which sets forth the driving problem to be solved. We then describe details of simulations that were carried out to analyze the framework and conclude by giving a schema of the framework architecture.

Experimental Setup

The driving problem for our experiments is to study the process of opinion formation driven by the following task: Train Dartmouth employees to convince the public that Dartmouth is a great school. To prepare our testbed, 2013 US News rankings data were used with attributes shown in Table 1. Each attribute is classified as binary with 0 representing a low score, and 1, a high score. We chose binary attributes to ensure simplicity in state space descriptions.

Table 1: US News 2013 College Ranking Data Attributes

Attributes	**Value = 0**	**Value = 1**
Ranking	≥ 100	≤ 100
Enrollment	≥ 20,000	≤ 20,000
SAT Scores	≤ 1,010	≥ 1,010
Graduation Rate	≤ 71%	≥ 71%
Class Size < 20	≤ 47	≥ 47%
Acceptance rate	≥ 35%	≤ 35

Using these attributes, random samples of schools were created as shown in Table 2. Three types of agents exist: trainers - advocates of the college; trainees - college recruiters; and testers - prospective students and parents. Trainees are trained by trainers to convince the public that Dartmouth is great. Each agent is defined by a goal profile based on a malleability - idealism scale, and passivity-activism scale. The malleability-idealism (γ) scale shows an agent's willingness to change its opinion (0 - lowest, 1 - highest), and the passivity-activism scale (ς) shows an agent's willingness engage in an interaction (0 - lowest, 1 - highest). Multiple types of agents are generated based on different combinations of (γ, ς). For our simulations, the primary focus was on malleable active (MA) agents (γ=1, ς= 0) who prioritize entirely on reaching consensus, and idealistic active (IA) agents (γ=1, ς= 1) who equally prioritize on maintaining original opinions and reaching consensus.

Table 2: University Feature Vector Samples derived from US Rankings 2013 data

Institution/Attribute	**1**	**2**	**3**	**4**	**5**	**6**
Harvard	1	0	1	1	1	1
UCLA	1	0	1	1	0	1
Wichita	0	1	0	0	1	0
UTEP	0	0	0	0	0	0
Brown	1	1	1	1	1	1

Using Table 2, feature vectors representing each school are constructed to form the world knowledge, K_w. Initially, all agents possess a subset of K_w. Since we have 5 schools in Table 2, $K_w = \{k_1; k_2; k_3; k_4; k_5\}$ where K_i is the i^{th} school. From K_w, we derive 32 different combinations of feature vectors of size 0 - 5 that an agent can initially possess in its knowledge base.

During an interaction, agents exchange information that is represented in form of queries. A query is represented as a vector with known and unknown values. Agents infer values of the unknowns to form an opinion. An example of a query might be: given that Dartmouth has an acceptance rate of 10%, is Dartmouth a highly ranked college? In this case, the query is represented as $q = [?, ?, ?, ?, ?, 1]$ with 5 unknown attributes. For a two person interaction, agents (i and j) exchange queries guided by their respective goals. At

each time step, each agent decides an action that maximizes its reward defined as:

$$R = -\gamma_i \ |O_i^t - O_j^t| \ - \ \varsigma_i |0_i^{\{t+1\}} - O_i^t|. \qquad (8)$$

O_i^t is agent i's opinion at time t, γ_i is agent i's malleability-idealism scale, and ς_i is agent i's passivity-activism scale.

After receiving a query, an agent has 3 types of cognitive actions that it can use to form a learning episode, L, which is the inferred opinion at that time. These actions are intadd, intremove, and donothing. For intadd, the agent updates its knowledge base by adding the current learning episode such that $K = K_{old} + L$. For intremove, the agent updates its knowledge base by removing the oldest learning episode and adding the current learning episode, $K = K_{old} + L - L_{old}$. For donothing, the agent maintains its previous opinion, $K = K_{old}$. To infer an opinion for each query, we applied a simple reasoning algorithm for inference instead of Bayesian Knowledge Bases (Yu and Santos 2016) which were used in the baseline framework. The algorithm infers values of unknown attributes (x) in a query vector by finding the mean of the prior values of that attribute in the agent's knowledge base (K) as shown in Equation 9:

$$E[\sum_{v \in K} attribute_x] \qquad (9)$$

Our reasoning behind this inference is the familiarity heuristic which states that an agent chooses an action that is usually familiar to its prior experiences (Metcalfe, Schwartz, and Joaquim 1993). Familiarity in our case is the frequency at which an attribute has been seen for similar types of objects by the agent. An example is as follows: Suppose an agent has only been exposed to 5 animals which include a lion, dog, hyena, giraffe, and a bird. If the agent is given a question to identify the diet of a newly discovered four-legged animal, the agent is likely to assume that the animal is carnivorous since the agent has seen mostly carnivorous four-legged animals. The same applies to our problem; if an agent has previously been exposed to highly selective schools, he is more likely to associate a new school with attributes of highly selective schools.

Simulations

For each interaction phase or episode, agents exchanged queries until they reached consensus or reached a maximum of 20 query exchanges without consensus. For each simulation, the trainee used some form of learned policy whilst the trainers/testers were guided by their goal profiles. In order to learn how to maximize reward, each trainee utilized an MDP with a state space defined by the cross product of the trainee and tester's opinions at time t, (O_1, O_2). Each opinion was defined by 6 binary attributes such that the state space spanned from (0 – 4095). All terminal states were defined such that $O_1 = O_2$. For baseline simulations, trainees were trained by trainers via q-learning to optimize their behavior based on their goal profiles. After training, the trainees underwent a testing phase where they utilized the learned reward function for their decision-making.

For the new framework, trainees undergo a learning phase, where they interact with testers for a defined number of episodes to generate trajectories which are fed to a DTM. The DTM generates a dynamic MDP which the agent uses for IRL to create a reward function. This function is used in the next q-learning phase to generate a policy for the agent. The process is constantly updated to help the agent adapt to its dynamic environment. A higher level architecture of the framework is summarized in Figure 1.

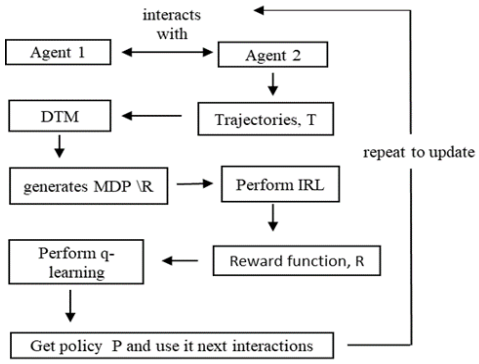

Figure 1: New Framework Architecture

Results

To evaluate the proposed framework, we assessed our success in addressing the original goals:

1. Can we capture the contextual ground truth to which agent preferences are made?

To answer this question, we carried out simulations using the c-neighbor and the maxent IRL algorithms. We chose the maxent method since it finds a solution that theoretically guarantees accuracy (Arora and Doshi 2018). The maxent algorithm assumes that the reward function is a linear combination of features, hence for comparison, we tested the c-neighbor algorithm which avoids this assumption.

Our assessment metric was to check if trajectories obtained from IRL policies were close enough to the expert policy. We generated 4 types of trajectories namely: T_{expert} - expert trajectories obtained after q-learning, T_{cneigh} - trajectories obtained from c-neighbor IRL policies, T_{maxent} - trajectories obtained from maxent IRL policies, and $T_{nopolicy}$ - trajectories obtained using the agent's goal profile without any learning. This is equivalent to performing q-learning at zero horizons. $T_{nopolicy}$ trajectories were designed as a worst case control measure to determine if IRL policies were meaningful. 200 simulations were run for each policy and agent types were equally incorporated form the set: {IAIA (IA-trainees, IA-testers), MAMA, MAIA, and IAMA}

To assess the performance of the obtained policies, average discounted feature expectations of all trajectories from each policy were computed as shown in Equation 10 (Abbeel 2004).

$$f_{exp} = E[\sum_{t=0}^{\infty} \gamma^t * f(s_t, a_t) | s \in \tau,\ a \in A, \pi] \quad (10)$$

$f(s_t, a_t)$ is the feature of a state s under an action a at time t, γ is the discount factor, and π is the policy. Deviations of each policy's average feature expectation from the expert demonstrations were calculated and results are in Table 4.

Table 4: Average deviations of feature expectations of obtained policies to the expert policy

Trajectories	cneigh	maxent	no-policy
20	2.70	1.62	1.79
50	2.18	1.21	1.52
100	1.47	0.99	1.53
500	1.12	0.82	1.56

Since our state space consisted of binary features on 6 attributes for 2 agents, the possible range of deviations varied from 0 to 3.46 ($\sqrt{(2*6)}$). As seen in Table 4, T_{maxent} yielded the lowest deviations compared to other policies. Low deviations meant that the expert feature expectations were close enough to the derived policy. This implies that with the maxent IRL policy, you would traverse through paths with features that were close enough to what the expert saw. The c-neighbor algorithm performed relatively well as the number of trajectories increased, but poorly under few trajectories. With fewer trajectories, the c-neighbor method considered only seen states and couldn't compute rewards for unseen states. As the number of trajectories increased, the algorithm improved in performance as more states were considered.

Since the maxent reward function was expressed in form of state vectors, we compared it to the ground truth which was in the same form. We could not do this comparison with the c-neighbor algorithm since its reward function was in form of state action triples. We computed the ground truth using Equation 8 for all states in the environment. Results obtained are shown in Figure 2. We can visually see that the maxent algorithm managed to match the distribution of ground truth rewards in terms of peaks and shape. The maxent policies yielded better results compared to the c-neighbor algorithm since the state space was small and finite. The linear combination of features assumption did not significantly affect accuracy since we utilized a simple familiarity inference algorithm that ignored non-linearity when the agent was undecided in inferring a value. With BKBs, non-linearity of features would be factored since these structures incorporate incompleteness in reasoning (Santos and Santos 1996). The maxent assumption would weakly hold due to this non-linearity.

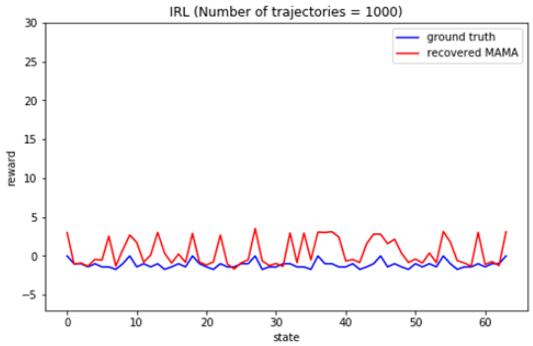

Figure 2: Maxent IRL vs ground truth reward function

Since the maxent algorithm gave the best matching based on feature expectations, we decided to use it for additional experiments in (2). To address our goal, we see that both IRL techniques were successful in estimating the preferences that agents were trying to optimize even though their goal profiles were unknown beforehand. However, the recovered reward functions had some error associated with them.

2. Can our framework adapt to dynamic situations, and reduce the problem of improper training?

Improper training can be defined as a situation when the expected experience differs from the observed experience. If an agent was only trained with MA agents who are agreeable, it will adopt policies that match MA behavior. However, in testing interactions, the agent might interact with other types of agents other than MA such that its learned policy will likely fail.

To test Question 2 for baseline simulations, 1000 episodes of interactions were ran for each q-learning phase and 2000 for the testing phases. For our new IRL framework, 500 episodes were allocated for each learning phase (maxent IRL and q-learning), and 2000 for testing in each update phase. For all testers, we gave them some randomness in their goal profiles such that they would randomly deviate from their normal behavior at a rate of 20% during interactions. This models realistic situations, were humans sometimes exhibit other characteristics that differ from their normal behavior. A predominantly compliant individual often shows some elements of stubbornness in certain situations.

We applied the update phase to modify the reward function as the interaction proceeded. Obtained results are shown in Table 5.

Table 5: Effect of dynamic conditions on consensus time

Training Type	mean consensus time	consensus time stdev
None	16.55	0.395
Baseline	15.91	0.231
IRL - 1 update	15.10	0.347
IRL - 2 updates	14.67	0.645
IRL - 3 updates	13.19	0.194

As seen from Table 5, IRL out-performed q-learning in capturing the changing behaviors of testers. In q- learning, the training was based on the assumption that an agent's interlocutor would behave consistently but during testing, the interlocutor exhibited some unknown behavior. This shows that q-learning based frameworks are susceptible to improper training if the testing conditions are not anticipated. Our IRL solves this issue by modelling preferences from past behavior. By revising the reward function, the framework improves in performance since the agent is kept up to date with of the changing behaviors of interlocutors.

Conclusion and Future Work

In this paper, we presented a framework that successfully combines both reinforcement learning and inverse reinforcement learning to model opinion dynamics. It models conditions that are likely to be successful in real life situations where an agent does not know its environment beforehand. Rather than relying on pre-crafted static parameters, which could be incomplete, the agent learns its preferences via IRL to get a more thorough picture of its preferences as seen from past behavior. Our framework captures the dynamism of opinion formation through reward updating to ensure that the agent captures its own changing attitudes, as well as other dynamic factors in the environment. The novelty of our framework lies in its ability to effectively learn based on how the interaction goes, whilst prior reinforcement learning frameworks learn based on what they expect.

However, our framework still presents some challenges for future work. The first challenge is that IRL methods usually require a large number of input trajectories to function well. This challenge makes it difficult to apply these frameworks in real life situations. More research needs to be done to find ways of improving performance using fewer trajectories. We would also like to scale the c-neighbor algorithm so that it can generalize to unseen states. Lastly, we plan to test our framework on human agents to assess its performance in real life scenarios and find ways to improve the architecture though better knowledge representation and application of more efficient heuristics.

Acknowledgements

This work supported in part by ONR Grant No. N00014– 1–2154, AFOSR Grant No. FA9550–15–1–0383, and DURIP Grant No. N00014–15–1–2514.

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.1-8.

Arora, S. and Doshi, P., 2018. A Survey of Inverse Reinforcement Learning: Challenges, Methods and Progress. arXiv preprint arXiv:1806.06877.

Damer, Edward, T. 2008. Attacking Faulty Reasoning: a Practical Guide to Fallacy-Free Arguments. Wadsworth Cengage Learning.

DeGroot, M. H. 1974. Reaching a consensus. Journal of the American Statistical Association. 69(345):118-121.

Ding Fei.; Liu, Yun.; Li, Yong. 2009. Coevolution of opinion and sttrategy in persuasion dynamics: An evolutionary game theoretical approach. International Journal of Modern Physics: 20(3): 479-490.

Friedkin, N.E. and Johnsen, E.C. 1990. Social influence and opinions. Journal of Mathematical Sociology, 15(3-4):193-206.

Gu, Q.; Santos, J.E. Santos, E.E. 2013. Modeling opinion dynamics in a social network. In Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT).2:9-16.

Kuhne, R. 2014. Political news, emotions, and opinion formation: Toward a model of emotional framing effects. In Annual Conference of the International Communication Association (ICA), Phoenix, AZ.

Lenz, G. S. 2009. Learning and Opinion Change, Not Priming: Reconsidering the Priming Hypothesis. American Journal of Political Science: 53(4): 821–837, 2009.

Metcalfe, J., Schwartz, B.L. and Joaquim, S.G., 1993. The cue-familiarity heuristic in metacognition. Journal of Experimental Psychology: Learning, Memory, and Cognition, 19(4):851-861.

Russell, S. 1998. Learning agents for uncertain environments. In Proceedings of the eleventh annual conference on Computational learning theory, 101-103.

Santos, Eugene, Jr., Nguyen, Hien, Kim, KeumJoo, Russell, Jacob A., Hyde, Gregory M., Veenhuis, Luke J., Boparai, Ramnjit S., De Guelle, Luke T., and Mac, Hung Vu. 2018. A Contextual Decision-Making Framework. to appear in Computational Context (Eds. W. Lawless), CRC Press.

Santos Jr, E. and Santos, E., 1996. Bayesian Knowledge-Bases (No. AFIT/EN/TR96-05). AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOOL OF ENGINEERING.

Sutton, R.S, and Barto, A.G. 1998. Introduction to reinforcement learning. Cambridge: MIT press.

Watkins, C.J, and Dayan, P. 1992. Q-learning. Machine learning, 8(3-4):279-292.

Yu, Fei. 2013. A Framework of Computational Opinions. PhD Diss. Dartmouth College.

Yu, F. and Santos, E., 2016. On Modeling the Interplay Between Opinion Change and Formation. In FLAIRS Conference, 140-145.

Ziebart, B.D. Maas, A.L. Bagnell, J.A. and Dey, A.K. 2008. Maximum Entropy Inverse Reinforcement Learning. In AAAI.8.1433-1438