Fast Abductive Reasoning Over Ontologies

Dan Davenport and Ray Hill
Lockheed Martin Advanced Technology Laboratories
3 Executive Campus, 6th Floor
Cherry Hill, NJ 08002
{ddavenport, rhill}@atl.lmco.com

Abstract

In this paper we present a new method for reasoning abductively over instances of a triples ontology. We compute the usefulness of evidence toward making an inference rather than its truth value, probabilistic or otherwise. This allows us to process ambiguous, incomplete, and inconsistent information effectively while remaining faithful to the ontology. The method is fast, scalable, and robust. We first motivate the method with a simple cognitive model and then present details of the algorithms. Finally, we present results from experiments that indicate that the method scales well in space and time complexity and promises to be highly effective.

Introduction

When analyzing alternative interpretations of data, we often reason abductively to decide which alternatives are most suitable for their purposes. Reasoning from an effect to a plausible cause is one form of abduction (Pierce 1903). For example, if I see a dent in my car door, I generally abduce that the cause is carelessness on the part of someone who parked next to me. Since other causes are possible, even if unlikely, we cannot say that this conclusion is deduced in the strict logical sense.

Proposals for implementing abductive reasoning abound (as do theoretical and philosophical treatments). Some are probabilistic, such as modified Bayesian methods (Keppens and Shen 2004), others are formal, such as abductive logic programming (Denecker and Kakas 2000), and some are hybrid (Poole 1993). Automated abduction has been used to integrate ontologies (Firat, Madnick, and Grosof 2002) and to integrate databases (Arieli, Denecker, Van Nuffelen, and Bruynooghe 2004). For an excellent overview of applications and methods see (Menzies 1996).

We first motivate our method with a simple cognitive model and then present details of the algorithms. Finally, we present results from experiments that indicate that the method scales well in space and time complexity and promises to be highly effective.

A Cognitive Model of Abduction

Our method of automated abduction is patterned after a "cognitive" model we have developed. That is, we took some hints from the way people seem to find plausible interpretations of facts (leaving aside issues related to human bias). Humans tend to focus on evidence that is perceived to be important (that is, salient within a context) and ignore the rest. They measure the relative importance of evidence within a whole context of facts, experience, aims and interests, putting salient facts in a cognitive framework to make sense of it (Kohler 1992).

We model abduction on the basis of this observation. Bringing what is important to salience while suppressing the rest is integral to abduction. Our approach is different from most, if not all, current methods, since they evaluate the truth value of propositions in one form or other—formally, probabilistically, "fuzzily," etc. What is determined to be true—by proof, probability, fuzzy set inclusion, or belief—is what they consider to be important.

In our model, an observation is important to the degree that it helps us discriminate observations that support a hypothesis from those that support it less strongly. We call this degree its *discriminability*. When an observation is important enough to be brought to our attention, we call it *salient* and call its importance its *salience*. That is, when an observation's discriminability rises above a threshold, we consider it to be salient to the hypothesis.

Our method of abduction does not and should not replace existing effective methods. We do not claim that it gives *better* results than other methods. It is not at all clear that that will be the case in general. We expect, however, that because it yields reasonable results (discussed later) over massive data sets that it will be an excellent complement to these methods.

Reasoning methods that are less scalable can be made more effective by using them as a post process to our method. We can return sets of triples that are very small so that other methods have a much smaller decision space to reason over. Complementing other methods with ours, therefore, makes them *effectively* scalable, even if they are not themselves scalable.

A Definition of Abduction

Abduction is the form of inference by which we reason from observed effects to plausible causes of them. Alternately, it is the form of inference that reasons from evidence to a plausible explanation of, or hypothesis about, that evidence. We adopt the latter characterization for purposes of this paper.

To make clear what we mean, we make the following definitions.

- *Plausible*: A hypothesis is *plausible* when it is at least superficially worthy of belief, that is, worth considering it or making it salient.
- *Evidence*: A observation is *evidence* when it is determined to be salient to a hypothesis.

Plausibility is a matter of human judgement. Our method only gathers evidence that such an explanation exists in a set of triples related by coreference. We don't "gist" the evidence, although that could be done in post-processing.

Algorithm for Abduction

The inputs of our algorithm include an ontology, a set of observations in triples form and a set of triples, related by coreference (forming a coreference graph), representing our interests. It returns sets of salient hypotheses, which are again sets of triples related by coreference (coreference subgraphs):

```
compute_property_saliences( Ontology );
compute_property_saliences( Observation_set );

repeat {

   compute_object_saliences( Observation_set );

   for each ( Object in the Observation_set ) {

      if ( salience_of( Object ) > threshold )
         do_abduction_step( Object );

   }

} until (no more abductions can be made);

compute_hypotheses();
```

In the first step, we determine how well ontology properties help us discriminate between triples of interest and those not of interest. In the first loop, we assign these discriminability values to instances. Next, we apply an algorithm to identify which object instances are most salient to the interests of a user. In the inner loop, we reclassify each object with a high enough salience in a more specific class. Given these new classifications we recompute the salience of each object until no more reclassifications can be made. We could further improve efficiency by removing objects whose salience thresholds fell below a certain threshold. We devote a section to each one of these steps.

Definitions and Notation

To explain how to compute the functions in the algorithm above, we need to make some definitions and introduce some notation.

We use Greek symbols to denote ontology classes and Latin symbols for instances. For purposes of this paper, we interpret triples differently than usual. We interpret the triple $\alpha(\Phi,\Gamma)$ to mean that it *can* be true, or, equivalently, that it is not inconsistent with the rest of the ontology. We interpret an instance $p(A,B)$ of $\alpha(\Phi,\Gamma)$ to mean that it *might* be true provided $\alpha(\Phi,\Gamma)$ *can* be true. To this end, we add (either explicitly or computationally) virtual properties that are inherited by all classes that subsume Φ. We illustrate this in Figure 1. This fact is keystone in understanding our method.

The reason for introducing "can be true" as opposed to "is necessarily true" is that abduction deals with *evidence*, which is not always fact. Also, no observation can be called evidence unless it is consistent with the ontology at the very least. We can abduce nothing from it (or worse, everything from it) unless we modify it in some manner to be consistent. This is not inconsistent with the normal truth theoretic interpretation since if it is necessarily true (in a consistent theory) then it can be true.

Discriminability of Property Classes

Given an ontology, our aim is to assign values to properties in a way that helps us to discriminate between possible abductive inferences. For example, if our ontology states that dogs and seals are animals and that animals can bark, then we have a choice between abducing that a particular barking animal is a dog or a seal. One way to guide such a decision is to assign each choice a likelihood that an animal that barks actually is a seal or a dog, as is often done in statistically based abduction methods.

While this is not necessarily a bad choice, we opt instead to use uniform distributions representing the probability that an observation is *useful* for abducing that a barking animal is a dog or a seal. There are three reasons for this choice. First, it is typically very hard to determine likelihoods accurately because prior probabilities are often hard to measure and can be very different in different contexts. Estimating posterior probabilities can be equally problematic.

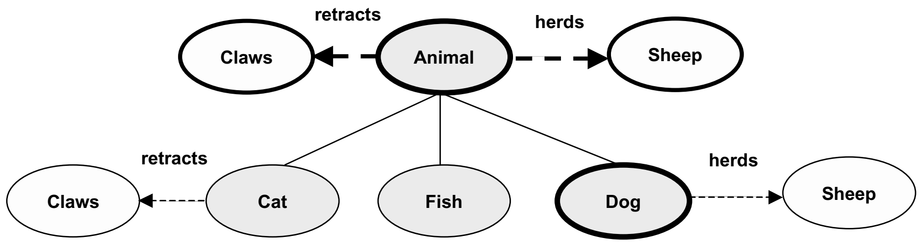

Figure 1. Virtual "Can Have" Properties

Second, choices that are most probable are not always the most useful. For example, if we are interested in evidence pointing to very rare phenomena, then we don't want to throw out unlikely instances just because they are improbable. Third, this choice is most consistent our definition of abduction in that we want to decide between choices that are plausible rather than choices that are merely probable.

For brevity, we say that an object class Φ is *subsumed directly* by another object Φ' if Φ' subsumes Φ, and Φ is maximal in the subsumption ordering with this property. Let Φ be subsumed directly by Φ', Γ subsumed by Γ' and $\alpha(\Phi',\Gamma')$. Define $\Pr(\alpha(\Phi',\Gamma'))$ to be the probability that Φ is marked given that an instance of Φ can have property α, that is, $\alpha(\Phi,\Gamma)$. We call $\Pr(\alpha(\Phi',\Gamma'))$ the *discriminability* of $\alpha(\Phi',\Gamma')$.

This is illustrated in Figure 2. At the top, the class triple "*Animal has_ear_color Color*" is assigned a discriminability of 0.5 since of the three classes directly under *Animal*, two have the property *has_ear_color* while only one is of interest. At the bottom all three have the required property so the class triple "*Animal has_nose_type Nose_type*" is assigned a discriminability of 0.33. To assign instance discriminabilities, each instance $a(A,B)$ in class $\alpha(\Phi',\Gamma')$ is assigned the discriminability of its class, $\alpha(\Phi',\Gamma')$.

Salience of Object Instances

In addition to assigning discriminability values to properties, we assign salience values to objects. Property discriminabilities are computed by considering how useful they are in discriminating between objects of interest and those not of interest. Our goal, however, is to find a plausible interpretation of the evidence. This being the case, we want to reinterpret triples representing general assertions to ones representing more specific assertions consistently with the whole of the evidence.

This is consistent with our definition of abduction since it is plausible that an animal is a dog, say, given sufficient evidence. In addition, we can recast the problem in terms of causality and say that the cause of the observations surrounding a putative animal are actually observations about a dog. So, given a body of evidence, we want to know when we should take an abduction step. That is, we need to determine how salient an object should be before we consider it for inferring that it has a more specific but plausible interpretation.

Although we compute property saliences by defining a probability distribution explicitly, we can't do the same for object saliences. We can, however, compute a set of probability values that are consistent with each other and with the property saliences. We do this even though we don't know what their distribution is *a priori*, thereby taking a Bayesian approach. We could alternatively define "units" of salience to create a distribution, but we have no real "physical" basis for doing so.

We define $\Pr(A)$ to be the probability that an instance A is *useful* in making an abduction within a specific context (such as deciding that a particular animal is actually a dog). As salience value gets larger, the plausibility that it is "abducible" grows. The trick is to compute this value from the salience values of properties and other objects and then compute the salience of these other objects from it in a recursive manner. We do this by computing a fixed point recursively for the whole set of object saliences. We thus obtain a self consistent set of probabilities that indicate whether to make an object salient to the user.

Suppose that A has positive salience, and $a(A,B)$ is discriminative. It makes sense that we should compute the salience of B as a function of both, since they can each affect its salience. If either is small, then the total effect on B should be small. If both are large the total effect should be large. It is not unreasonable, therefore, to define the effect of both on B to be their product. In addition, any one of the properties of A could have an effect on its salience, so we combine them in a kind of salience sum.

We compute the salience of objects using the discriminability of their properties and the salience of their domains. We define $\Pr(A_i)$ to be: $1-\prod_j \left(1-\Pr(a_j(A,B_j))\Pr(B)\right) = 1-\prod_j \left(1-\delta_j \sigma_j\right)$ where δ_j is a property discriminability and σ_j an object salience. This function has the properties we require, although other definitions are possible. It also makes it easy and efficient to compute at large scales.

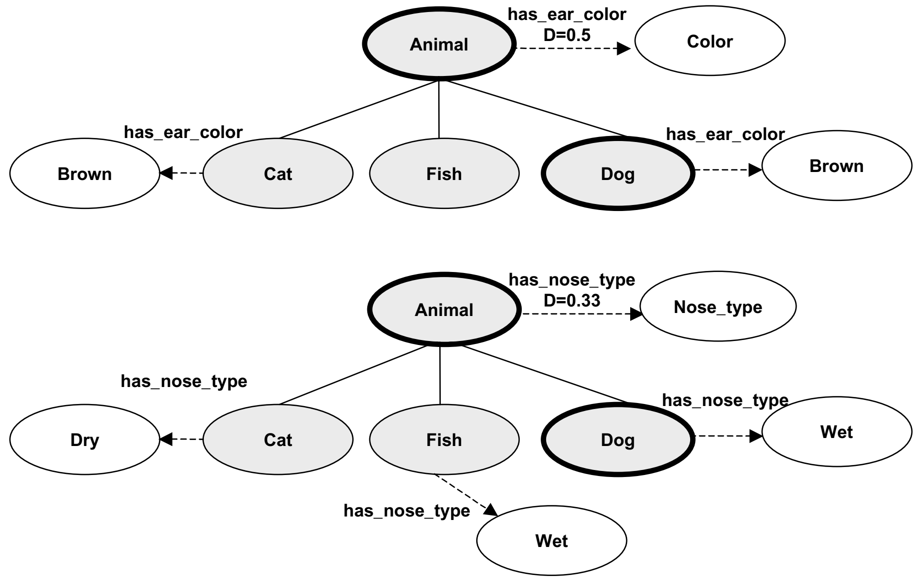

Figure 2. Computing the Discriminability of Property Classes

Suppose we have a collection of instances with discriminability $\Pr\big(a(A_i, B)\big)$. We can also define the discriminability of the inverse $a^{\tau}(B, A_i)$ of $a(A_i, B)$ implicit in the observation $a(A_i, B)$. For each *i* we call $\Pr\big(a^{\tau}(B, A_i)\big)$ the *dual* of $\Pr\big(a(A_i, B)\big)$. We compute each of these values separately and use each to compute distinct saliences for each object, one as the range of a property and one as the domain.

Define $\delta_{ij} = \Pr(a_{ij}(A_i, B_j))$, $\delta_{ji} = \Pr(a_{ji}(B_j, A_i))$, $\tau_i = \Pr(A_i)$ and $\sigma_j = \Pr(B_j)$. We can then write a pair of dual equations $\sigma_i = 1 - \prod_j \big(1 - \delta_{ij}\tau_j\big)$ and $\tau_i = 1 - \prod_j \big(1 - \delta_{ji}\sigma_j\big)$.

Efficient Computation of Object Saliences

Define an operator $\oplus$ (the $\oplus$-sum) on the unit interval by $u \oplus v = u + v - uv$ (sometimes used as a "fuzzy or" operator). It's easy to show that $\oplus$ is commutative and associative and that $1 - \prod_j (1 - u_j) = \bigoplus_j u_j$. We can, therefore, rewrite the equations above as $\sigma_i = \bigoplus_j \delta_{ij}\tau_j$ and $\tau_i = \bigoplus_j \delta_{ji}\sigma_j$.

Formally, this looks like multiplying a vector by a matrix with addition replaced by $\oplus$. Define the matrix Δ by $\Delta_{ij} = \delta_{ij}$. We can rewrite the equations in the matrix form $\sigma = \Delta^t[\tau]$ and $\tau = \Delta[\sigma]$, where the square brackets denote our modified form of matrix multiplication and where Δ^t is the transpose of Δ.

Solving the matrix equations simultaneously we get two fixed points that give us a set of object saliences, two per object, that are self-consistent. This is formally similar to Kleinberg's algorithm for determining the value of web pages (Kleinberg 1998) but very different in that we are not computing the principle components of two matrices simultaneously, but rather dual fixed points of a nonlinear system of equations. We've proven that the dual fixed points are unique and can be computed with exponential convergence. This corroborates our experience that the algorithm converges in seven to eleven steps depending on how many triples there are in the observation corpus.

Since $\oplus$ is associative, we can rewrite $\bigoplus_j u_j$ as $\bigoplus_j u_j = u_1 \oplus \big(\bigoplus_{j\neq 1} u_j\big) = u_1 + \big(\bigoplus_{j\neq 1} u\big)_j - u_1\big(\bigoplus_{j\neq 1} u_j\big)$. Hence we can compute a $\oplus$-sum cumulatively almost as fast as we can compute a normal row sum. We need to be careful, though, about round off errors for values near one.

Constructing Discriminability Matrices for Instances

Given a set of observations, which are instances of class triples, we use the discriminabilities of their property classes as their discriminabilities, or more precisely, the property of the triple. However, we need to put all these values into a matrix and its dual so we can compute salience values for their domains and ranges.

As a first step, and to illustrate the basic idea, we define a matrix for a single property, say "has_color." We create an $n \times m$ matrix where *n* is the number of distinct instances of animals on the observation corpus and *m* is the number of distinct colors in the corpus. For each instance of a *has_color* triple in the corpus ("*Animal has_ear_color Color*," for example) we assign the (i, j) entry of the matrix the discriminability of the triple. We compute the dual matrix in a similar manner but do so using the dual discriminabilities we computed for each dual class triple.

We can use these matrices as building blocks to create a single pair of "master" matrices that are used to compute object saliences. One simple way to do this is to create matrices for each relation. We then assign fill (i, j) with the point-wise $\oplus$-sum of the discriminabilities of all the properties between objects i and j for matrices of the same dimensions. We did this in the experiments mentioned below.

This does not, however, take advantage of all the information we can glean from user specification of what is of interest to them. This method also ignores assertions that are contradictory and assertions that are not consistent with the ontology's rules. In addition, we have not mentioned how ontology rules factor into our method. However, we have developed strategies to address these issues and have some useful results, which we will report in later communications.

Abduction Step

Suppose that we have an aggregation of observations, one of which involves an animal (see Figure 2). We see that, of the three animal types in our ontology, only one is of interest (thick border). Our goal is to accumulate enough evidence that the animal in question has brown ears (through relations with other evidence) to plausibly infer that it is a dog, since dogs are what we are interested in. We then reclassify the object in the observation as a dog.

Having done this, we change the discriminability values of its (observed) properties, using those values computed for the properties of the new classification (dog). We then recompute object salience values until no saliences are above the threshold.

Notice that after we make an abduction, we may be able to deduce more information not found directly in the evidence. For example, we may have a rule that asserts that dogs owned by people are domesticated. As long as such deductions consume a reasonable amount of resources, our algorithm will still scale.

Salience of Hypotheses

To find and rank hypotheses, which are collections of evidence, we search breadth first within the coreference graph from seed objects. The seed objects are chosen by $\oplus$-summing their (dual) salience values and then thresholding. We call this sum the *evidential value* of an object. We stop the search at objects whose evidential value is smaller than a membership threshold. We then rank these collections by the $\oplus$-sum of their individual evidential values.

Quality of Results

In one experiment we measured the quality of our results by subjectively assessing the meaningfulness of working groups within the social structure of the Lockheed Martin Advanced Technology Laboratories. We collected 44,000 email messages sent between 200 employees over a one month period. Consequently the density of links to nodes in the communication graph was highly dense, making it very difficult to find meaningful subgraphs representing working groups. Figure 3 illustrates one of several working groups detected, the "command structure" of LM ATL. Another revealed the existence of three proposal teams coordinated by a single person and his manager.

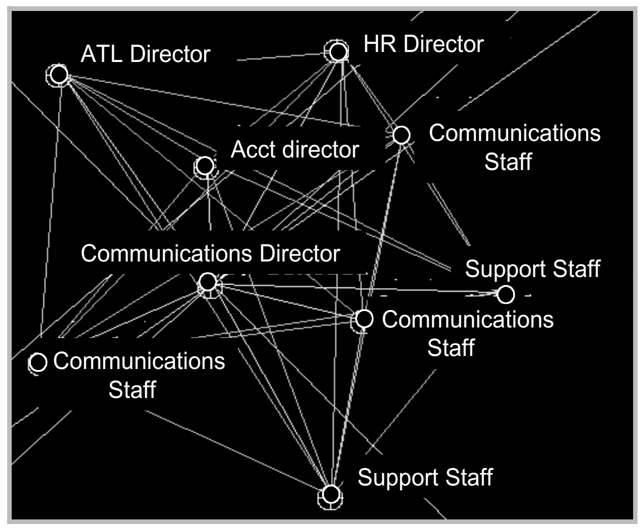

Figure 3. "Command" Structure of LM ATL

We only used header information for privacy purposes, encoding each person as a number. One person familiar with LM ATL personnel evaluated the results to find groups that were meaningful. We used a flat "ontology" consisting of the email properties *To:*, *From:*, *CC:* and *BCC:*.

To derive salience values for email properties we counted the number of emails between individuals with the same property class and normalized the counts. We gave each person the salience value computed by $1-\prod(1-r)^n = \bigoplus_i^n r$ where n was the number of emails sent between two individuals and $r = 0.1$ was an arbitrary value we assigned to each email. Since $\oplus$-summing is monotone, any number between 0.1 and 0.9 would have sufficed. Beyond these limits round-off error becomes a significant factor.

Scalability

Our method scales linearly when the triples used for input are power law distributed (alternately self scaling, or having small world statistics). This is because the expected value of links per node is constant. For many important data sets this seems to be the case (Barabasi 2003).

The data we used in the scaling experiment were graciously supplied by the program manager of the Eagle project funded by ARDA. The corpus we used consisted of about 1.2 million links with about 2,000 nodes (people), around 200 of which were objects of interest. The scaling results are illustrated in Figure 4.

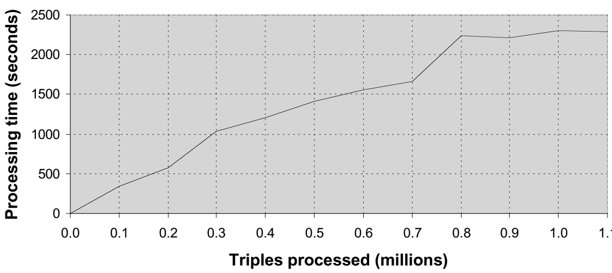

Figure 4. Scaling Results for Social Link Data

As we had predicted, our algorithm scaled linearly. Above about 800,000 links the curve flattens out, which we attribute to an artifact of the data set, which was artificially generated. We attribute the linearity to the fact that the data was approximately power law distributed, a common occurrence in link data (triples). Due to the size of the corpus, we did not conduct a subjective assessment of the quality of experimental results.

We've obtained the same linearity on two other link corpora as well provided by an LM ATL customer each of which contained over 400,000 links and 60,000 people. A subjective assessment of quality was conducted by the customer who expressed pleasure at the result.

References

Pierce, C. S. 1903. Abduction and Induction. In Buchler, J., ed. *Philosophical Writings of Pierce*, 150-156. New York, Dover Publications, Inc.

Keppens, J., and Shen, Q. 2004. Causality Enabled Compositional Modeling of Bayesian Networks. In Proceedings of the 18th International Workshop on Qualitative Reasoning, 33-40.

Denecker, M., and Kakas, A. C. 2000. Abductive Logic Programming. *A special issue of the Journal of Logic Programming*, 44(13).

Poole, D. 1993. Probabilistic Horn Abduction and Bayesian Networks. *Artificial Intelligence*, 64(1):81-129.

Firat, A. Madnick, S. and Grosof, B. 2002. Knowledge Integration to overcome Ontological Heterogeneity: Challenges from Financial Information Systems. In Proceedings of the ICIS-2002. 183-194. Barcelona, Spain.

Arieli, O., Denecker, M., Van Nuffelen, B., Bruynooghe, M. 2004. Coherent Integration of Databases by Abductive Logic Programming. *Journal Artificial Intelligence Research (JAIR)*, 21:245-286/

Menzies T. 1996. Applications of Abduction: Knowledge Level Modeling. *International Journal of Human Computer Studies.* 45:305-355.

Kohler, W. 1992. *Gestalt Psychology: An Introduction to New Concepts in Modern Psychology*. New York: Liveright Publishing Corporation.

Kleinberg, J. 1998. Authoritative sources in a hyperlinked environment. In Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms.

Barab´asi, A.-L. 2003. *Linked: How Everything is Connected to Everything Else and What It Means for Business, Science, and Everyday Life.* New York: Plume.