Transforming between Propositions and Features: Bridging the Gap

Daniel T. Halstead and Kenneth D. Forbus

Qualitative Reasoning Group
Northwestern University
1890 Maple Avenue, Evanston, IL 60201, USA
halstead@cs.northwestern.edu

Abstract

It is notoriously difficult to simultaneously deal with both probabilistic and structural representations in A.I., particularly because probability necessitates a uniform representation of the training examples. In this paper, we show how to build fully-specified probabilistic models from arbitrary propositional case descriptions about terrorist activities. Our method facilitates both reasoning and learning. Our solution is to use structural analogy to build probabilistic generalizations about those cases. We use these generalizations as a framework for mapping the structural representations, which are well-suited for reasoning, into features, which are well-suited for learning, and back again. Finally, we demonstrate how probabilistic generalizations are an excellent bridge for joining reasoning and learning by using them to perform a traditional machine learning technique, Bayesian network modeling, over arbitrarily high order structural data about terrorist actions, and further, we discuss how this might be used to facilitate automatic knowledge acquisition.

Introduction

Both propositional representations of data and feature-value representations are each useful for different tasks. "Flat" feature vectors are useful in representing uncertainty, and therefore are very appropriate for doing learning tasks. However, structured representations are more useful when dealing with the range of information that people learn and manipulate. Plans, explanations, and arguments are among the kinds of important information that often require explicit representation of relationships. Indeed, there is evidence suggesting that the use of language and analogy to accumulate relational knowledge is why humans are so smart compared to other animals (Gentner, 2003). Understanding how to learn the same breadth of knowledge, with the same flexibility and robustness as people do, requires understanding how relational learning works.

While both representations are useful, what is lacking is a simple way of converting from one to the other. Working with both structure and probability has proven to be quite difficult, as it is not clear how to combine the two. Probability requires a single, uniform representation of each case. That is, there must be a guarantee that the values for which one is aggregating probabilities correspond to the exact same feature or concept in every record or case. This guarantee is made implicitly when doing learning from flat, table-like data. However, when the data is represented as arbitrary structured assertions, this guarantee is much more difficult to make.

Analogy is an excellent tool for bridging this gap. By making an analogy, one is postulating that many of the assertions in one case correspond to the same or similar concept as many of the assertions in another case. As we will show, it is this notion that allows us to generate probabilistic information even when dealing with large amounts of high-order, highly-interconnected structural data.

In this paper, we demonstrate how it is possible, using analogy, to build fully specified probabilistic models from structural case descriptions about terrorist attacks. The process consists of two stages: generalization and flattening. We describe each of these stages in the next two sections. In the fourth section, we present an example of how the procedure was used to build a real probabilistic model. Finally, in the last two sections, we discuss related work and possible future directions.

Probabilistic Generalizations

Formalizing probability over expressions

It is often unclear exactly what probability should convey when dealing with relations of arbitrarily high order. Therefore, we first present a formal definition of how we use probability in these scenarios. For example, should the notion of probability for an expression like:

```
(EVENTOCCURSAT TERRORISTATTACK-1997-SRILANKA COLOMBO)
```

be different from the notion of probability for an expression like:

```
(THEREEXISTATLEAST 2 ?X
 (AND ((CITIZENSFN UNITEDSTATESOFAMERICA) ?X)
  (ORGANISMKILLED TERRORISTATTACK-1997-SRILANKA ?X)))?
```

In the first expression, it is not useful to talk about the probability of whether such an expression occurs in the case at all. In fact, if the case knowledge is complete, such an expression should occur in every case, since every terrorist attack should be tied to some location. Instead, it is much more useful to talk about the probability distribution of the various locations. We therefore create a random variable, 'Location', for this expression. The variable can take on different values over different cases.

Note that it is necessary to make two critical simplifying assumptions in order to do this: first, that there is a finite number of possible locations; and second, that each location is mutually exclusive. Neither assumption is strictly true. Locations can be arbitrarily well-specified (e.g., a city, a country, or a three foot radius), and an event can occur in multiple locations at once (e.g., the September 11th attacks in the United States). Since we only analyze the distribution of values that have been actually observed by the system during training, we do not worry about violating the first assumption as long as one value is not a specification of another. This introduces an experiential bias, but it has been shown that such a bias is often useful (Kahneman & Tversky, 1979). We discuss how to handle other violations of these *non-specificity* and *mutual exclusivity* assumptions later in the paper.

In the second expression however, it is most useful to talk simply about the probability of whether the fact is present in the case knowledge or not. Although it may be tempting to Curry this expression, so that we instead examine something like the probability distribution of the citizenship of the victims, note that in this context, citizenship is not mutually exclusive: there could well be both American and Sri Lankan victims of the attack. Therefore, for each possible citizenship, we would need to create a separate random variable to represent its existence. As long as this expression is matched only to other expressions describing American casualties during the analogical matching process (as our process does), this is identical to creating a random variable for the existence of the expression itself in a given case.

We thus have two separate notions of probability, each useful for different relations. We refer to the first as *characteristic probability* and the second as *existential probability*. During the generalization stage, we generate only existential probabilities. Then during the flattening stage, we use knowledge of the semantics of each predicate in a given expression to determine whether to partition these into characteristic probabilities instead.

Constructing Generalizations

We build probabilistic generalizations over the input case descriptions as a means of providing a uniform representation (aka relational schema) for all of the cases. This is done by using analogy to determine which concepts in one case best correspond to the concepts in another case. We can then use the probabilistic generalization as a framework for flattening the case descriptions down into features. This provides a natural way to easily transform descriptions from propositions into features and back again.

Our approach to analogy is based on Gentner's (1983) *structure-mapping theory* of analogy and similarity. In structure-mapping, analogy and similarity are defined in terms of structural alignment processes operating over structured representations. The output of this comparison process is one or more *mappings*, constituting a construal of how the two entities, situations, or concepts (called *base* and *target*) can be aligned. A mapping consists primarily of a set of *correspondences* and a *structural evaluation score.* A correspondence maps an item (entity or expression) from the base to an item in the target. The structural evaluation score indicates overall match quality.

We use the Structure-Mapping Engine (SME) to do analogical mapping and compute structural evaluation scores (Falkenhainer, Forbus, & Gentner, 1986). SME uses a greedy algorithm to compute approximately optimal mappings in polynomial time (Forbus & Oblinger, 1990). The base and target descriptions can be pre-stored, or they can be dynamically computed based on queries to a large knowledge base (Mostek, Forbus, & Meverden, 2000).

The generalization algorithm itself is a probabilistic implementation of SEQL (Kuehne, Forbus, et. al). SEQL is designed to produce generalizations incrementally from a stream of examples. It uses SME to compare each new example to a pool of prior generalizations and exemplars. If the new example is sufficiently close to one of the generalizations, it is assimilated into that generalization. Otherwise, if it is sufficiently close to one of the prior exemplars, it is combined with it to form a new generalization. By "sufficiently close", we mean that the structural evaluation score (after normalization, to account for differences in the size of descriptions) exceeds a pre-set threshold. In our probabilistic implementation, this process of constructing or extending generalizations is done by taking the union of the expressions in the two descriptions, and adjusting the probability of each according to whether or not it was in the overlap as found by SME. Matching entities that are identical are kept in the generalization, and non-identical entities are replaced by new entities that are still constrained by all of the statements about them in the union. Should no sufficiently close match be found, the example is simply added to the pool of exemplars.

More formally, given an initially empty set of generalizations G and set of exemplars X, a threshold σ, and a function Score(*x*,*y*) which calculates the similarity (i.e. the normalized structural evaluation score) between two descriptions, SEQL operates as follows:

1. Receive an exemplar x.
2. Look for a generalization y∈G s.t. Score(x,y) > σ. If one is found, proceed to step 5.
3. Look for an exemplar y∈X s.t. Score(x,y) > σ. If one is found, proceed to step 5.
4. Add x to X. Repeat from step 1.
5. Remove y and add Generalization(x,y) to G. Repeat from step 1.

There are two parameters for controlling the behavior of this algorithm. The first is σ, a threshold which the similarity between the base and target must surpass in order to be accepted and incorporated into a generalization. It is tempting, given our goals of creating a feature description of the whole domain, to set σ to be as low as possible, so that as much of the domain as possible will be incorporated into the generalization. However, in practice, we have found that a low σ actually makes the matches so poor that the features constructed from them are meaningless. A value of 0.9 for σ (out of a maximum of 1, since the similarity score is normalized) performs much better in this regard.

Secondly, when our generalizations grew to a very large size (including many infrequent statements), we have found it useful to invoke a probability cutoff ρ. This means that any statements with probability less than ρ are ignored during the primary match process for the sake of speed, efficiency, and to reduce the chance of over-fitting. However, they are kept around for a certain number of rounds, inversely proportional to the cutoff itself, in case they should ever rise above the cutoff again. When ρ is low, a great many expressions are included in the generalization which only occur a very few number of times throughout the case descriptions and there is a high risk of over-fitting the data; however, when is ρ high, we risk losing the significant low-frequency information that is crucial for feature discrimination. We have found a value of 0.2 to behave reasonably.

Features

The second stage of the feature-generation process is to use the probabilistic generalization as a frame for the actual flattening of the structural case representations into features and values. Features may need to represent existential or characteristic probability, and continuous or discrete values. Therefore, we distinguish between three different types of features: existential, characteristic, and continuous.

Feature Types

An *existential* feature is one which, like the second expression in section 2.1, or like the expression (BOMBING ATTACK1), is best described by an existential probability. An existential feature is TRUE whenever a matching expression occurs in the case. When there is no matching expression though, then there is a question of whether its value should be MISSING rather than FALSE. We label a feature as MISSING when its generalized expression contains any entity which is described by other statements in the case, but which has no corresponding entity in the analogy. For example, suppose a generalization makes many references to ATTACK1. If the expression (BOMBING ATTACK1) is missing from a case, then that feature will be false only if something corresponding to ATTACK1 is actually present somewhere else in the case description. This has the effect of reducing non-causal dependencies within the data, since otherwise the values of all features containing the same entity would be identical whenever that entity was not present.

In contrast, *characteristic* features encompass those expressions which it is more useful to describe by some characteristic probability over the distribution of one of their arguments. An example is the expression (EVENTOCCURSAT TERRORISTATTACK-1997-SRILANKA COLOMBO) from section 2.1. In our implementation, we treat an expression as a property if it contains one entity which is either: always a *constant* (e.g. 2); always a *function* (i.e. non-atomic); or, the only other entity in the expression is the case-entity (as in the example above). If there are two such arguments, we could create two such features; however, we have not yet found this to be necessary. This decision criteria is not perfect, but we have found it to work remarkably well in practice. The value of a characteristic feature can be either MISSING (when a matching expression is absent), or any of the observed values of the corresponding argument.

Finally, as its name implies, a *continuous* feature is simply a characteristic feature for which the possible values of the argument in the expression are continuous. An example is the expression (CASUALTYCOUNT TERRORISTATTACK-1997-SRILANKA 23). There are several ways to handle such a situation. Currently, we simply discretize the range (using k-means with 3 buckets). Thus, the possible values for a continuous property are MISSING, LOW, MEDIUM, or HIGH. Other possible solutions we could implement later if necessary are to use a probability density instead of a distribution, or to check for background knowledge about the significant intervals of the range, such as might be expressed in QP-theory (Forbus, 1984).

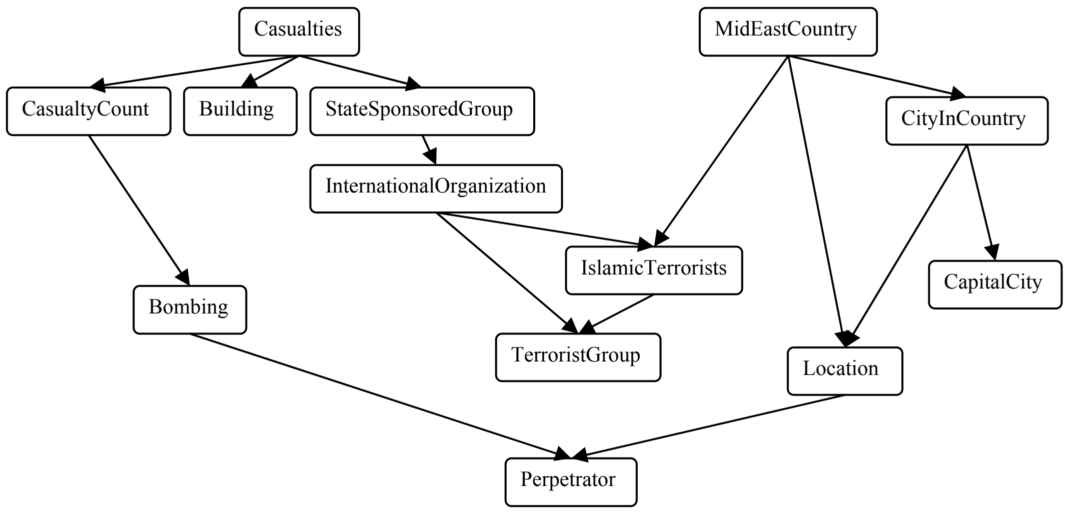

Figure 1. A Bayes net generated automatically from structural descriptions of terrorist attacks.

Flattening

After the probabilistic generalization has been built, we look at the possible values of each expression within to determine which of the four types of features it should correspond to. By storing the feature description and the generalized expression together, we construct an invertible mapping for each expression. When the mapping is applied to an expression, it will generate a feature-and-value representation, and vice-versa when its inverse is applied to a feature and value.

Finally, the mapping can be used to transform the propositions in the original case descriptions into features. We iterate through each expression in each of the original descriptions, compare it to the probabilistic generalization to determine which generalized expression it corresponds to, and then apply the corresponding feature mapping. In this way, we can build up a joint probability table of how the values of every feature co-vary across each case. We use an ad-tree (Anderson & Moore, 1998) to cache all of these joint probabilities.

There are two possible aforementioned problems with this approach. The first lies in violation of the *non-specificity* assumption. We have shown how propositions of arbitrarily high order can be transformed into feature representations. However, we would like to be able to say the same about entities of arbitrarily high specificity. Note that when the specificity is described in other facts outside of the expression itself, there is no problem. However, when the specificity is internal to the entity's representation, such as in `(CITYINCOUNTRYFN SRILANKA)` or `(DAYFN 11 (MONTHFN SEPTEMBER (YEARFN 2001)))`, then there is a matching problem: the extra structure will prevent the parent expression from matching the expressions in other cases that it should. This results in duplicated expressions in the generalization. We solve this by substituting a unique, generic zero-order symbol for the non-atomic entity. However, this introduces a new problem, because now information that was carried in the original non-atomic entity (such as the month and year) is lost. We reacquire it by asserting it outside of the expression, introducing a new predicate `TERMOFUNIT` for that purpose. We recursively apply more substitution where necessary. For example, if the symbol `:GENFN23` were substituted for the date function described above, then we would also assert in separate facts:

```
(TERMOFUNIT :GENFN23 (DAYFN 11 :GENFN24))
(TERMOFUNIT :GENFN24 (MONTHFN SEPTEMBER :GENFN25))
(TERMOFUNIT :GENFN25 (YEARFN 2001))
```

Secondly, there is the *mutual exclusivity* assumption. This one is less clear how to solve when it is violated. One possible solution is to create a second generalization, over the set of multiple instances. In the 9/11 example, we would create one generalization of all three attacks, and use the generalization as the input case. This has not yet been implemented.

Learning Example: A Bayesian Network

It is exciting to have a means for building a completely specified joint probability table over any set of arbitrarily interconnected, high-order structural data. From the joint probabilities, we can construct a large variety of probabilistic models. Here we show how the method enabled us to build a Bayesian network model from the propositional descriptions of terrorist attacks.

We used a basic, naïve Bayes approach to modeling the data. Although this requires making strong assumptions about independence, it did quite well in the domain we tested it under. Additionally, we hope that with the immense background knowledge that comes from large propositional KB's, and by having propositional representations of the data, we can in the future do a better job of recognizing where and how such independence assumptions are violated. If so, we could account for them and focus future learning efforts on those assertions which are not provably related. We computed the structure of the Bayes net using simulated annealing with random restarts. The scoring heuristic used in the hill-climber was a Bayes Information Criterion estimate (Friedman and Yakhini, 1996). It was trained over a corpus of 2,235 different descriptions of terrorist attacks, provided by Cycorp, and entered manually by domain experts. The descriptions ranged in size from 6 to 158 propositions, with the average being 20 propositions.

The diagram above shows a Bayes net that we generated using this procedure. These are preliminary results for which we have not yet consulted an expert opinion. However, it is an encouraging sign of validation to note that the features which describe location are all closely connected, as are the features which describe the terrorist group.

Related Work

Most past methods for doing learning from symbolic evidence amount to approaches in the style of evidential reasoning (Pearl, 1987). They require a causal model to start with, and then infer probabilities based on observations and simulation from the model. In contrast, we try to both infer the probabilities and induce the model from the evidence alone. We also are set apart from newer approaches which do try to do this in that we set no preconditions about either the uniformity of input representations, or the order of the expressions in those representations.

The approach most similar to our own algorithm is that of Getoor et al. (2001), of probabilistic relational models or PRMs. A PRM is a Bayesian dependence model of the uncertainty across properties of objects of certain classes and the relations between those objects. It extends traditional Bayesian networks to incorporate a much richer relational structure. It can also handle information aggregated across several members of a class within the same case (for example, a student's highest grade or the lowest age among the victims of an attack).

However, the PRM approach is limited in two ways: first, it can only model first-order relations; and second, it has trouble when there is no prior knowledge of a *relational schema* or uniform representation of each case. Although several papers have been published to try to overcome this second limitation (the modeling of *existence variables* (Getoor, et al., 2002) is particularly similar to our approach), none seems to present a uniform syntax for overcoming all forms of structural uncertainty, and none includes a method for modeling higher order relations. By contrast, our approach uses independently validated cognitive models of analogical matching to build such a unifying relational schema, from arbitrary predicate calculus descriptions of arbitrarily high order. A hybrid approach might be promising, using a PRM built upon the probabilistic generalizations we construct to provide the necessary schema.

Several other new approaches have recently arisen in the field of relational learning for doing model induction. For example, Blockeel and Uwents (2004) present a method for building a neural network from relational data, and Dayanik and Nevill-Manning (2004) discuss clustering relational data through a graph-partitioning approach. Recent work in link analysis also provides a means for tying probability in with relational data to do learning. Cohn and Hofmann (2001) actually create joint probability tables of both properties and links (in this case, terms and citations in document analysis) through a probabilistic decomposition process related to LSA, and then use it to perform classification. Variants on ILPs such as Bayesian logic programs (BLPs) (Kersting, de Raedt, & Kramer, 2000) have also been suggested for this sort of application. However, to our knowledge, there is not yet any approach which can fully satisfy the two conditions we have stated: learning relations of arbitrarily high order, and learning without any knowledge of prior relational schema.

Other related work includes Keppens and Shen (2004), who demonstrate a way to build a Bayesian network from process knowledge such as that expressed by qualitative process theory (Forbus, 1984). Tenenbaum & Griffiths (2001) provide a very well conceived and conveyed description of alternative models for generalization, all derived from Bayesian hypothesis formulation.

Finally, a number of alternative cognitive simulations of analogical mapping and retrieval have been developed. Some are domain-specific (Mitchell, 1993), and thus inapplicable to this problem. Others are based on connectionist architectures (Hummel & Holyoak, 1997; Eliasmith & Thagard, 2001), and are known to not be able to scale up to the size of examples used in these experiments. While CBR systems have been used to tackle similar structured representation problems (Leake, 1996), they also tend to be built as special-purpose systems for each domain and task. By contrast, the simulations used here are applicable to a broad variety of representations and tasks.

Future Work

Although we can now build probabilistic models from arbitrary propositional data, these are only preliminary results. A great deal remains to be done to improve the process. Properly dealing with violations of the mutual exclusivity assumption is probably of foremost importance. It would also be useful to modify the generalization process, so that it drops out expressions with low information gain rather than those with low probability.

Outside of improving the process itself, there are two directions we intend to explore in the future. The first of these concerns exploring the application of this procedure to other problems and other models. For example, it might be interesting in some domains to build a decision tree rather than a Bayes net. Also, exploring the application of the Bayes net to knowledge acquisition should prove interesting. For example, one can imagine a scenario in which each edge in the Bayes net is a conjectured relationship that is tested against background knowledge in the KB. Those conjectures which are consistent with the KB provide validation. Those that are inconsistent indicate either an error in the inputs or a poor model. Those which are neither represent new conjectures which might be beneficially explored in order to learn more about the domain.

Also, it would be highly interesting to further explore how the tools of probability can help in symbolic reasoning, and vice-versa. For example, there are many roles that statistical measures such as information gain and significance might play in symbolic inference. In the other direction, propositional background knowledge might help us to determine when assumptions about independence are being violated, so that learning efforts can be adjusted accordingly.

Acknowledgements

This research was supported by the Information Processing Technology Office of the Defense Advanced Research Projects Agency. All opinions presented here are those of the authors, not those of the U.S. government.

References

Anderson, B., & Moore, A. (1998). ADTrees for Fast Counting and for Fast Learning of Association Rules. *Knowledge Discovery from Databases*, New York, 1998.

Blockeel, H., & Uwents, W. (2004). Using neural networks for relational learning. *ICML-2004 Workshop on Statistical Relational Learning and its Connection to Other Fields,* pp.23-28.

Cohn, D., & Hofmann, T. (2001). The missing link – a probabilistic model of document content and hypertext connectivity. *Advances in Neural Information Processing Systems 13:430-436,* MIT Press.

Dayanik, A. and Nevill-Manning, C. G. (2004). Clustering in Relational Biological Data. *ICML-2004 Workshop on Statistical Relational Learning and Connections to Other Fields*, pp. 42-47

Eliasmith, C. and Thagard, P. 2001. Integrating structure and meaning: A distributed model of connectionist mapping. *Cognitive Science*.

Falkenhainer, B., Forbus, K. and Gentner, D. The Structure-Mapping Engine. *Proceedings of the Fifth National Conference on Artificial Intelligence.* 1986.

Forbus, K. (1984). Qualitative process theory. *Artificial Intelligence*, 24, 85-168

Forbus, K. Exploring analogy in the large. In Gentner, D., Holyoak, K., and Kokinov, B. (Eds.) *Analogy: Perspectives from Cognitive Science.* MIT Press. 2001.

Gentner, D., Structure-mapping: A theoretical framework for analogy, *Cognitive Science* 7(2), 1983

Gentner, D. (2003). Why we're so smart. In D. Gentner and S. Goldin-Meadow (Eds.), *Language in mind: Advances in the study of language and thought* (pp.195-235). Cambridge, MA: MIT Press.

Getoor, L., Friedman, N., Koller, D., & Taskar, B. (2002). Learning probabilistic models of link structure. *JMLR* 3, 679-707.

Getoor, L., Friedman, N., Koller, D., & Pfeffer, A. Learning probabilistic relational models. In Dzeoski, S. and Lavrac, N. (Eds.), *Relational Data Mining* (pp. 307-335). Kluwer, 2001.

Hummel, J.E., & Holyoak, K.J. (1997). Distributed representations of structure: A theory of analogical access and mapping. Psychological Review, 104.

Kahneman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica, 47,* 263–292.

Keppens, J., & Shen, Q. (2004). Causality Enabled Modeling of Bayesian Networks. *Proceedings of the 18th International Workshop on Qualitative Reasoning*, 33-40.

Kersting, K., de Raedt, L., & Kramer, S. (2000). Interpreting Bayesian logic programs. *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pp. 29-35.

Kuehne, S. E., Gentner, D. & Forbus, K. D. (2000). Modeling infant learning via symbolic structural alignment. *Proceedings of the Twenty-second Annual Conference of the Cognitive Science Society*, 286-291.

Leake, D. (Ed.) 1996. *Case-based Reasoning: Experiences, Lessons and Future Directions*, MIT Press.

Mitchell, M. (1993) *Analogy-making as perception: A computer model.* MIT Press, Cambridge, MA.

Mostek, T., Forbus, K. and Meverden, C. 2000. Dynamic case creation and expansion for analogical reasoning. Proceedings of AAAI-2000. Austin, Texas.

Tenenbaum, J., & Griffiths, T. (2001). Generalization, similarity, and Bayesian inference. *Behavioral and Brain Sciences* 24 (629-640).