Research Statement

Brian Milch

My primary areas of interest are artificial intelligence and machine learning. Over the last two decades, these fields have been revolutionized by probabilistic graphical models: models such as Bayesian networks and Markov random fields, which compactly describe the probabilistic dependencies among large sets of variables. Algorithms for learning and reasoning with graphical models now support a wide range of practical applications, from predicting the functions of proteins to helping autonomous vehicles determine where they can drive safely.

But graphical models are not the last word in probabilistic AI. For instance, a graphical model learned from a pediatric medical database could include many links between individual variables — Richard tends to get sick when Laura is sick, Miguel tends to get sick when Maria is sick, etc. — but it could not express an overall pattern such as "children tend to catch diseases from their siblings". The need to represent patterns that hold across whole classes of variables has led to the development of *relational* probabilistic models. These models combine the probabilistic semantics of graphical models with the ability of first-order logic to generalize across objects. This fusion of logic and probability is a promising way forward for AI — but it raises formidable challenges for knowledge representation, inference, and learning. My research focuses on these challenges.

Current Work

My Ph.D. thesis introduced a relational probabilistic language called Bayesian logic, or BLOG, which goes beyond earlier languages in modeling scenarios with unknown objects. I will begin by discussing BLOG's representational contributions. Then I will describe an approach to probabilistic inference based on Markov chain Monte Carlo over "partial worlds", with an application to disambiguating bibliographic citations. Finally, I will describe my post-doctoral work on Bayesian structure learning for relational probabilistic models.

Probabilistic models with unknown objects. Consider tracking vehicles using multiple cameras, or making inferences about businesses based on reviews from several web sites. In such scenarios, we have no authoritative list of the vehicles or businesses we want to reason about; we are uncertain about how our observations match up with underlying objects. This problem of "data association", "record linkage" or "entity resolution" is ubiquitous, and special-purpose algorithms have been developed for dealing with it. But because scenarios with unknown objects are not easily reduced to a fixed set of variables with fixed dependencies, it has remained difficult to address them using the general-purpose toolkit of graphical models. Even relational probabilistic languages usually assume that the objects in any given scenario are known in advance.

Working with colleagues at U.C. Berkeley, I developed Bayesian logic (BLOG), a relational probabilistic language that can easily describe scenarios with or without unknown objects. A BLOG model defines a probability distribution over "possible worlds" that include varying sets of objects with varying relations among them. To define a distribution on this rich outcome space, the model compactly specifies conditional distributions for two kinds of basic variables: variables indicating the number of objects that satisfy certain properties, and variables indicating the value of a function or predicate on some arguments. Although such a model can be seen as an infinite Bayesian network, it is outside the scope of previous results ensuring that such networks fully define a distribution, because it may contain cycles and some nodes may have infinitely many ancestors. In a paper at AISTATS 2005, we introduced a new theory of *contingent* Bayesian networks (CBNs). CBNs explicitly represent the context-specific nature of dependencies: for instance, the color of a given pixel depends on the color of vehicle 92 only if that pixel is in an image segment corresponding to that vehicle. In an IJCAI 2005 paper, we used this theory to give conditions under which a BLOG model is "well-formed" and fully defines a distribution over possible worlds.

BLOG has been implemented in an inference engine that is available online. The engine includes a top-down sampling algorithm that can answer queries on any well-formed BLOG model. Although the model may define infinitely many variables, this algorithm samples only those variables that are context-specifically relevant given the values sampled so far. The engine also provides a Markov chain Monte Carlo framework that can be customized for particular applications, as I will describe below. Prof. Josh Tenenbaum has used BLOG to teach a computational cognitive science class at MIT, and several students at U.C. Irvine have used it for class projects.

Markov chain Monte Carlo over partial worlds. Answering queries on a relational probability model is often intractable, particularly in scenarios with unknown objects. One technique that has proven successful is Markov chain Monte Carlo (MCMC), where we execute a random walk over possible worlds and record the distribution of query values in the worlds we visit. MCMC can be customized by using a domain-specific proposal distribution, which guides the sampler toward high-probability worlds. In a paper at NIPS 2002, we used this approach to reconstruct the set of distinct publications referred to by a set of highly varied and sometimes erroneous citations.

Until recently, using MCMC with a customized proposal distribution has required implementing not just the proposer itself, but also code to compute the probabilities of worlds and accept or reject the proposed moves. In a UAI 2006 paper with Stuart Russell, I described an MCMC inference engine for BLOG that eliminates much of this work. The user provides a BLOG model and a proposal distribution; generic code does the rest. In addition to implementation issues, this work involved a theoretical challenge: the proposal distributions used in practice do not propose fully-specified worlds. For instance, a proposal might set the total number of papers in a field to 1000, but propose titles and author names only for the 200 papers that it connects to observed citations. This proposed "partial world" can be thought of as describing a whole set of particular worlds. I derived conditions under which MCMC over such partial worlds can be performed efficiently and yields correct answers to queries. We evaluated this inference framework using a citation dataset obtained from CiteSeer, showing that it obtains the same accuracy as an earlier hand-coded implementation and is slower by only a small constant factor.

Bayesian structure learning for relational probabilistic models. Each relational probabilistic language has a grammar for defining dependency models. In a UAI 2007 paper with Ashwin Deshpande and other MIT colleagues, I explored a Bayesian approach to learning accurate dependency models for a given data set. We worked with a language for describing how the world changes as a result of an agent's actions. For instance, if a robot in a "blocks world" scenario executes a `pick-up` action on a block, then there is some probability that the block ends up in the robot's gripper, but some probability that it is just knocked onto the table. The dynamics of the world is modeled by a set of probabilistic action-effect rules. Each rule includes both a first-order logical component specifying when it applies, and a probability distribution over possible action effects.

Taking a Bayesian approach, we defined a prior distribution over rule sets — in fact, since we also wanted to transfer knowledge between related environments, we used a hierarchical model including both environment-specific rule sets and a global "rule set prototype". The Bayesian approach naturally penalizes more complex rule sets by assigning them lower prior probabilities. Our learning algorithm performs a greedy local search for the rule sets with the greatest posterior probability given the data. Some of our search operators are driven by the data: for instance, they add rules to cover particular training examples. This combination of a Bayesian objective function, a local search algorithm, and data-driven search moves yielded impressive empirical results. The whole transfer learning algorithm ran in less than 10 minutes, and achieved high accuracy with significantly fewer examples than a non-transfer learner on a variety of blocks-world domains.

Future Work

My work in AI is motivated by our field's long-term goal of building general-purpose systems that learn to reason about everything, from cooking to carpentry to cosmology. Such a system could answer such diverse queries as "How can I get from Boston to New Haven without a car?", "How many members of the U.S. Congress have doctoral degrees?", and "Find me an appropriately sized case for this digital camera". Coding the necessary knowledge by hand seems hopeless; instead, the system would need to learn a model of the world from online text documents, images, and videos. Although we are quite far from implementing such a system, my research agenda will be guided by asking what stands in the way of doing so.

One area I will focus on is efficient inference. Although we now have modeling languages that combine probability and logic, we are only beginning to unify the algorithms used for probabilistic and logical reasoning. There has been some cross-fertilization: techniques based on weighted model-counting are being used for probabilistic inference, and probabilistic belief propagation algorithms have inspired a successful SAT algorithm called survey propagation. But it is not clear how to extend SAT-based algorithms to perform approximate inference, and to reason about continuous variables. I will investigate these issues over the next few years. A group of us at MIT are also working on lifted inference algorithms, which exploit symmetries in a model by abstracting over interchangeable objects. A particularly exciting idea is to exploit *approximate* symmetries, ignoring small differences in beliefs about different objects in order to maintain tractability.

I will also focus on structure learning for relational probabilistic models. The BLOG language defines a rich class of relational dependency models; I will work on algorithms for searching this model space efficiently. Based on our experience with learning action-effect rules (as well as the literature on inductive logic programming), it appears that bottom-up or example-driven search techniques hold a lot of promise. However, I would also like to move beyond learning dependencies among a fixed set of relations. A learning system should be able to hypothesize new relations and object types: for instance, it might explain co-authorship patterns in a publication database by hypothesizing a "knows" relation between researchers, or inventing "research group" objects.

Such predicate invention techniques could be applied to limited versions of the web-based learning problem that I discussed above. For example, customers evaluate different features for different kinds of products — battery life for electronics, warmth for gloves, etc. — and it would be exciting to discover these features automatically from online reviews. By developing new algorithms for learning and inference in relational probabilistic models, I plan to expand the range of useful tasks that computers can perform without domain-specific programming.