Toward Statistical Predicate Invention

Stanley Kok KOKS@CS.WASHINGTON.EDU
Pedro Domingos PEDROD@CS.WASHINGTON.EDU
Department of Computer Science & Engineering, University of Washington, Seattle, WA 98195, USA

1. Motivation

In the past few years, the statistical relational learning (SRL) community has recognized the importance of combining the strengths of statistical learning and relational learning (also known as inductive logic programming (ILP)), and developed several novel representations, as well as algorithms to learn their parameters and structure. However, the problem of statistical predicate invention (SPI) has so far received little attention in the community. SPI is the discovery of new concepts, properties and relations from data, expressed in terms of the observable ones, using statistical techniques to guide the process and explicitly representing the uncertainty in the discovered predicates. These can in turn be used as a basis for discovering new predicates, which is potentially much more powerful than learning based on a fixed set of simple primitives. Essentially all the concepts used by humans can be viewed as invented predicates, with many levels of discovery between them and the sensory percepts they are ultimately based on.

In statistical learning, this problem is known as *hidden* or *latent* variable discovery, and in relational learning as predicate invention. Both hidden variable discovery and predicate invention are considered quite important in their respective communities, but are also very difficult, with limited progress to date.

One might question the need for SPI, arguing that structure learning is sufficient. Such a question can also be directed at hidden variable discovery and predicate invention, and their benefits, as articulated by their respective communities, also apply to SPI. The benefits of SPI over structure learning include the following:

1. SPI gives a more compact and comprehensible model of a data-generating process than pure structure learning. An invented predicate efficiently captures dependencies among observed predicates. Instead of directly modeling these, which could require an exponential number of parameters, we can introduce an invented predicate and model the dependence between it and each of the observed predicates, which requires only a linear number of parameters. We can view the invented predicate as a summary of the information in the observed predicates, and the conduit through which the information is relayed to other observed predicates. For example, in citation domains, a standard observed predicate is $\texttt{Author}(\texttt{person}, \texttt{paper})$. We could invent a predicate $\texttt{Coauthor}(\texttt{person}, \texttt{person})$ representing the concept of coauthorship, defined as $\texttt{Author}(\texttt{x}, \texttt{z}) \wedge \texttt{Author}(\texttt{y}, \texttt{z}) \Rightarrow \texttt{Coauthor}(\texttt{x}, \texttt{y})$. With the invented predicate, we can more compactly learn (weighted) formulas such as $\texttt{Affiliation}(\texttt{x}) \wedge \texttt{Coauthor}(\texttt{x}, \texttt{y}) \Rightarrow \texttt{Affiliation}(\texttt{y})$. By combining features of both statistical and relational learning, SRL is more complex than either. Thus SPI's ability to produce a less complex, more compact model is even more important in SRL than in statistical learning and relational learning.
2. By having a more compact model, we have fewer parameters, thereby reducing the risk of overfitting. A more compact model also reduces the amount of memory required to represent the model, and could potentially speed up inference.
3. Once a predicate has been invented, it can be used to learn new formulas, potentially allowing us to take larger steps through the search space, and learning more complex models than would otherwise be possible.
4. With invented predicates, we can represent unobserved aspects of a data-generating process, and learn a better model of it. This could potentially improve the accuracy of the learned model.

Potential applications of SPI include:

- **Perception.** In object recognition in visual scenes, we would like to invent predicates representing pixel configurations that correspond to parts of objects, and predicates representing objects as related sets of parts. Similar considerations apply to speech recognition, handwriting recognition, etc.
- **Molecular Biology.** Computational biology has so far focused mainly on predicting properties of individual molecules (e.g., identifying promoter regions in DNA, or predicting the secondary structure of proteins). However, the outcomes that biologists and medical researchers ultimately care about involve the interactions of many such molecules (Hood & Galas, 2003). With SPI, we can learn predicates representing gene modules (i.e., co-regulated sets of genes), metabolic pathways, substructures of the cell, etc.
- **Security.** Criminal activities like money laundering or the preparation of a terrorist attack can often not be detected from a single event, but only by a complex pattern of events that may at first appear unrelated (Jensen & Goldberg, 1998). SPI enables us to learn predicates representing the steps of a criminal's plan, the relations

Presented at the ICML Workshop on Open Problems in Statistical Relational Learning, Pittsburgh, PA, 2006.

among them, the roles played by different individuals, etc.

2. State of the Art

Elidan, Friedman and coworkers have developed a series of algorithms for finding variables in Bayesian networks. Elidan et al. (2001) look for structural patterns in the network that suggest the presence of hidden variables. Elidan and Friedman (2005) group observed variables by their mutual information, and create a hidden variable for each group. Central to both approaches is some form of EM algorithm that iteratively creates hidden variables, hypothesizes their values, and learns the parameters of the resulting Bayesian network. A weakness of such statistical approaches is that they assume that the data is independently and identically distributed, which is not true in many real-world applications.

In relational learning, the problem is known as predicate invention (see Kramer (1995) for a survey). Predicates are invented to compress a first-order theory, or to facilitate the learning of first-order formulas. Relational learning employs several techniques for predicate invention. Predicates can be invented by analyzing first-order formulas, and forming a predicate to represent either their commonalities (interconstruction (Wogulis & Langley, 1989)) or their differences (intraconstruction (Muggleton & Buntine, 1988)). A weakness of inter/intraconstruction is that they are prone to over-generating predicates, many of which are not useful. Predicates can also be invented by instantiating second-order templates (Silverstein & Pazzani, 1991), or to represent exceptions to learned rules (Srinivasan et al., 1992). Relational predicate invention approaches suffer from a limited ability to handle noisy data.

Only a few approaches to date combine elements of statistical and relational predicate invention. Popescul and Ungar (2004) apply k-means clustering to the objects of each type in a domain, create predicates to represent the clusters, and learn relations among them. Perlich and Provost (2003) present a number of approaches for aggregating multi-relational data (e.g., aggregating over the fields of a single table and across tables). We would like SPI to automatically and selectively invent predicates corresponding to useful clusters and aggregates. Craven and Slattery (2001) proposed a learning mechanism for hypertext domains in which class predictions produced by naive Bayes are added to an ILP system (FOIL) as invented predicates. We would like SPI to be a general-purpose mechanism that can be used across all domains, not only hypertext. Davis et al. (2005) use an off-the-shelf ILP system to learn Horn clauses on a mammogram database. They create a predicate for each clause learned, add it as a feature to the database, and then run a standard Bayesian network structure learning algorithm. Rather than treating predicate invention and structure learning as two separate sequential steps, we would like SPI to closely integrate the two.

3. One Approach

We are currently developing an approach to SPI that combines elements of hidden variable discovery and ILP predicate invention. Inspired by Elidan and Friedman (2005), we group observed predicates that are highly correlated with each other, and invent a predicate for each group. To do so, we measure the correlations of all pairs of predicates (considering all variabilizations in which the predicates share at least one variable), and discard pairs with low correlation. The result can be viewed as a graph, with predicates as nodes, and arcs representing correlations. We then find approximate, possibly overlapping cliques in the graph, and invent a predicate for each clique. We model the correlation among the predicates in the group by adding a weighted edge between the invented predicate and each of its observed predicates. The arguments of the invented predicate are (a subset of) the arguments of the observed ones. The weight of an edge reflects the amount of correlation between an observed predicate and all the others in the group. We then repeat the process with the new predicates added to the pool, thus allowing multiple levels of predicates to be invented.

References

Craven, M., & Slattery, S. (2001). Relational learning with statistical predicate invention: Better models for hypertext. *Machine Learning*, *43*, 97–119.

Davis, J., Burnside, E., Dutra, I., Page, D., Ramakrishnan, R., Costa, V. S., & Shavlik, J. (2005). View learning for statistical relational learning: With an application to mammography. *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*. Edinburgh, Scotland.

Elidan, G., & Friedman, N. (2005). Learning hidden variable networks: The information bottleneck approach. *Journal of Machine Learning Research*, *6*, 81–127.

Elidan, G., Lotner, N., Friedman, N., & Koller, D. (2001). Discovering hidden variables: A structure-based approach. *Advances in Neural Information Processing Systems 14* (pp. 479–485). Cambridge, MA: MIT Press.

Hood, L., & Galas, D. (2003). The digital code of DNA. *Nature*, *421*, 444–448.

Jensen, D., & Goldberg, H. (Eds.). (1998). *Proceedings of 1998 AAAI Fall Symposium on Artificial Intelligence and Link Analysis*. Orlando, FL: AAAI Press.

Kramer, S. (1995). *Predicate invention: A comprehensive view* (Technical Report). Austrian Research Institute for Artificial Intelligence, Vienna, Austria.

Muggleton, S., & Buntine, W. (1988). Machine invention of first-order predicates by inverting resolution. *Proceedings of the Fifth International Conference on Machine Learning* (pp. 339–352). Ann Arbor, MI: Morgan Kaufmann.

Perlich, C., & Provost, F. (2003). Aggregation-based feature invention and relational concept classes. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 167–176). Washington, DC: ACM Press.

Popescul, A., & Ungar, L. H. (2004). Cluster-based concept invention for statistical relational learning. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 665–664). Seattle, WA: ACM Press.

Silverstein, G., & Pazzani, M. J. (1991). Relational clichés: Constraining constructive induction during relational learning. *Proceedings of the Eighth International Workshop on Machine Learning* (pp. 203–207). Evanston, IL: Morgan Kaufmann.

Srinivasan, A., Muggleton, S. H., & Bain, M. (1992). Distinguishing exceptions from noise in non-monotonic learning. *Proceedings of the Second International Workshop on Inductive Logic Programming (ILP'92)* (pp. 97–107). Tokyo, Japan.

Wogulis, J., & Langley, P. (1989). Improving efficiency by learning intermediate concepts. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 657–662). Los Altos, CA: Morgan Kaufmann.