# Probabilistic-Logic Models: Reasoning and Learning with Relational Structures

Manfred Jaeger
*Institut for Datalogi, Aalborg Universitet,*
*Selma Lagerlöfs Vej 300, DK-9220 Aalborg Ø*
jaeger@cs.aau.dk

## Introduction

Over the last decade several strands of research in Artificial Intelligence and Machine Learning have come together in an emergent field sometimes called *probabilistic logic learning* or *statistical relational learning*. In this extended abstract the origins, development and some current challenges of this field are briefly sketched.

## Probability and Logic in AI

Knowledge representation and reasoning under uncertainty is one of the long-standing challenges for AI. In most approaches to reasoning under uncertainty, the classical calculus of probabilities is used as the underlying framework for quantifying uncertainty. To implement probabilistic reasoning in a formal system, the first natural idea was to build on standard logics, and extend their syntax and semantics so as to obtain systems in which one could not only reason about the truth of falsity of a proposition, but more generally about the probability of a proposition being true. Propositional logic was extended in this way by Nilsson [11] (and, in fact, already 130 years earlier by Boole [2]); first-order logic by Halpern [5] and Bacchus [1].

Several problems emerged for using these *probabilistic logics* in practice: the computational complexity of probabilistic inference, the weak implications often obtained in these logics (i.e. a knowledge base *KB* would often not entail much more for a query proposition $\phi$ than that the probability of $\phi$ lies between 0 and 1), and the fact that the logic-based representation languages were not well suited to express knowledge about stochastic independence or causal relations, two important aspects of probabilistic reasoning.

As a result, *probabilistic graphical models*, notably *Bayesian Networks* [12,9], became the more successful paradigm for probabilistic reasoning in AI. Graphical models specify a unique distribution over the possible worlds (and hence over the propositions) for a fixed propositional vocabulary. The specification of graphical models relies heavily on knowledge of independence and/or causality, and inference in graphical models, while still intractable in the worst case, has proven to be feasible in many practical applications.

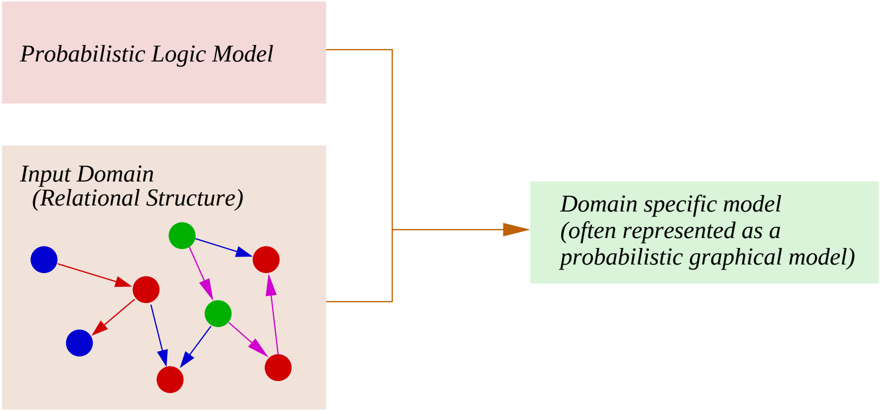

**Figure 1.** Probabilistic Logic Models

There is a price to pay for these advantages of graphical models over logic-based representations: first, graphical models require a high specification effort, and do not allow for a modular, incremental compilation of partial (probabilistic) knowledge. Second, a probabilistic graphical model is restricted to one particular domain represented by its propositional variables, and does not allow to express more high-level knowledge that generalizes over wide classes of domains. For example, a single graphical model could represent the probability distribution over the propositional variables *bloodtype(John,A)*, *bloodtype(John,B)*, *bloodtype(John,AB)*, *bloodtype(John,0)*. . . , *bloodtype(Mary,A)*, . . . , *bloodtype(Mary,0)*, . . . *bloodtype(Paul,0)*, representing the bloodtype in the domain *John*, *Mary*, *Paul*, where, say *Mary* and *Paul* are the parents of *John*. However, we can not represent a general model about the probabilities of bloodtypes and the laws of inheritance that could be applied to arbitrary pedigrees.

The second limitation of graphical models is addressed in frameworks for *knowledge based model construction* [3]. Here high-level representation languages using elements of first-order logic are used to specify general knowledge that for each concrete domain (consisting of a set of objects, and possibly some known structure, e.g. the kinship relations in a pedigree) defines a unique probability distribution over a domain-specific set of propositional variables. We call any such high-level model a *probabilistic-logic model*, Figure 1. Examples of formal languages for the specification of probabilistic-logic models are *Prism* [16], *relational Bayesian networks* [6], *Bayesian logic programs* [10], and *Markov logic networks* [15].

## Learning from Structured Data

The classical data model in machine learning consists of a list of *examples* (or observations, data-items, . . . ), each of which consists of values for a certain set of attributes. However, in many modern applications of machine learning, the available data is not easily represented in this format: the world wide web in web mining, bio-molecular data in bioinformatics, social network data – data often comes in the form of labeled graphs, trees, time sequences, or, specifically, relational databases, rather than a plain attributevalue table. Specialized sub-fields of machine learning, notably *inductive logic programming* [13] and *graph mining* [4] have long considered such non-standard forms of data.

Probabilistic-logic models afford a unifying view of many different types of data and connect some of these traditional disciplines of machine learning: many non-standard forms of data can be seen as models over a finite domain of objects (web pages, atoms, ...) for a logical language containing relation symbols of various arities representing attributes and relations (links between web pages, bonds between atoms, ...) of objects. Structured data, thus, has the form of input domains for a probabilistic-logic model as depicted in Figure 1, and Probabilistic-logic models can be used as predictive models for structured data. For example, a probabilistic-logic model for a certain genetic trait over pedigree input domains can be used to predict whether a given person is affected by that trait. Moreover, predictive models with structured output fall within the scope of probabilistic-logic models: a probabilistic-logic model, for example, can also return a probability distribution over possible kinship structures, given an input domain specified only by a set of persons and some of their (genetic) attributes. Thus, the model could be used to predict the underlying pedigree structure from observed genetic data.

## Relational Bayesian Networks

Relational Bayesian Networks [6,7] are a representation language for probabilistic-logic models that is based on the syntax of *probability formulas*. These formulas can be seen as probabilistic generalizations of predicate logic formulas: a predicate logic formula $\phi(x_1, \ldots, x_k)$, built via the syntactic constructors *atomic formulas*, *boolean connectives*, and *quantification*, defines for every tuple $c_1, \ldots, c_k$ of domain elements a truth value $\phi(c_1, \ldots, c_k) \in \{\mathit{true}, \mathit{false}\}$. A probability formula $F(x_1, \ldots, x_k)$, built via the syntactic constructors *atomic formulas*, *convex combinations*, and *combination functions* (which closely correspond to the three predicate logic constructors), defines for every tuple $c_1, \ldots, c_k$ of domain elements a probability value $F(c_1, \ldots, c_k) \in [0, 1]$.

A main strength of the Relational Bayesian Network language is the parsimony and recursive nature of its syntax, which enables theoretical analyses as well as algorithmic procedures to be performed by a straightforward induction over the construction of probability formulas. For example, the learning from data of parameter values in relational Bayesian networks is essentially performed by computing partial derivatives of probability formulas by induction over their syntactic form [8].

The *Primula* system (http://www.cs.aau.dk/∼jaeger/Primula) is a publicly available implementation of relational Bayesian networks.

## Challenges

From a Computer Science and AI perspective, one of the main benefits of developing and studying probabilistic-logic languages is to identify the common, abstract structure of probabilistic models for many domains, and to provide uniform inference and learning methods that are applicable over a wide spectrum of domains and application types. However, it is often unrealistic to expect that the generic algorithms implemented in a system like *Primula* can compete against highly engineered special purpose tools for concrete application tasks like protein structure prediction, or other core tasks in bioinformatics. Nevertheless, the flexibility and richness of probabilistic-logic modeling languages affords also for such applications new opportunities for developing and testing new types of predictive models. The potential of probabilistic-logic models in bioinformatics applications has been demonstrated in a major EU research project, which is documented in [14]. The more specific application for biological sequence analysis is the subject of an ongoing Danish national research project (http://lost.ruc.dk).

Many challenges also remain in the further theoretical and algorithmic development of probabilistic-logic modeling. With regard to inference problems, the focus, so far, has been on "fixed domain" inference problems: how to compute probabilities in models induced by one concrete input domain. However, one can also consider more general questions like: what are the bounds for the probability values of a certain proposition that are obtained when the model is instantiated over a range of input domains? Can inference results obtained for one domain be dynamically updated under incremental changes of the domain?

## References

[1] F. Bacchus. *Representing and Reasoning With Probabilistic Knowledge.* MIT Press, 1990.

[2] G. Boole. *Investigations of Laws of Thought on which are Founded the Mathematical Theories of Logic and Probabilities*. London, 1854.

[3] J. S. Breese, R. P. Goldman, and M. P. Wellman. Introduction to the special section on knowledge-based construction of probabilistic decision models. *IEEE Transactions on Systems, Man, and Cybernetics*, 24(11), 1994.

[4] D. J. Cook and L. B. Holder, editors. *Mining Graph Data*. Wiley, 2007.

[5] J.Y. Halpern. An analysis of first-order logics of probability. *Artificial Intelligence*, 46:311–350, 1990.

[6] M. Jaeger. Relational bayesian networks. In Dan Geiger and Prakash Pundalik Shenoy, editors, *Proceedings of the 13th Conference of Uncertainty in Artificial Intelligence (UAI-13)*, pages 266–273, Providence, USA, 1997. Morgan Kaufmann.

[7] M. Jaeger. Complex probabilistic modeling with recursive relational Bayesian networks. *Annals of Mathematics and Artificial Intelligence*, 32:179–220, 2001.

[8] M. Jaeger. Parameter learning for relational Bayesian networks. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007.

[9] F.V. Jensen and T. D. Nielsen. *Bayesian Networks and Decision Graphs*. Springer, 2007.

[10] K. Kersting and L. De Raedt. Towards combining inductive logic programming and bayesian networks. In *Proceedings of the Eleventh International Conference on Inductive Logic Programming (ILP-2001)*, Springer Lecture Notes in AI 2157, 2001.

[11] N. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28:71–88, 1986.

[12] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. The Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, San Mateo, CA, rev. 2nd pr. edition, 1988.

[13] L. De Raedt. *From Inductive Logic Programming to Multi-Relational Data Mining*. Springer, 2008.

[14] L. De Raedt, P. Frasconi, K. Kersting, and S.H. Muggleton, editors. *Probabilistic Inductive Logic Programming*, volume 4911 of *Lecture Notes in Artificial Intelligence*. Springer, 2008.

[15] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107 – 136, 2006.

[16] T. Sato. A statistical learning method for logic programs with distribution semantics. In *Proceedings of the 12th International Conference on Logic Programming (ICLP'95)*, pages 715–729, 1995.