Expressivity Analysis for PL-Languages

Manfred Jaeger JAEGER@CS.AAU.DK
Aalborg University, Fredrik Bajers Vej 7E, DK-9220 Aalborg

Kristian Kersting KERSTING@INFORMATIK.UNI-FREIBURG.DE
Luc De Raedt DERAEDT@INFORMATIK.UNI-FREIBURG.DE
University of Freiburg, Georges-Koehler-Allee, Building 079, D-79110 Freiburg

Abstract

We propose a framework for analyzing the expressivity of probabilistic logical languages.

1. Introduction

Many different languages for representing probabilistic-logical (pl) models have been proposed over the last decade. Among them are Prism models [9], Stochastic Logic Programs (SLPs) [6], Relational Bayesian networks (RBNs) [3], Probabilistic Relation Models (PRMs) [1], Bayesian Logic Programs (BLPs) [5], Markov Logic Networks (MLNs) [8], and many others.

There now is considerable interest in gaining a better understanding of the relationships between these languages, and translations between them have been investigated [7, 10, 4, 2]. These works focus on pairwise comparisons between languages and do not establish a general, robust framework for comparing pl-languages. In this paper we propose the outlines of a general conceptual framework for analysing expressivity of pl-languages.

Our approach is based on several premises: first, we take the position that an expressivity comparison should be grounded in a common semantic framework for the languages under investigation, so that expressivity can be formalized by the ability of a language to capture certain models. Second, we propose that the comparison should focus on relatively basic versions of the different languages. They will typically embody the particular strengths and weaknesses of a representation paradigm more clearly than language variants that have been extended in various ways. The point to bear in mind is that a result "L is less expressive than L'" may well be more interesting than a result "L is at least as expressive as L'", because it may very well be accompanied by a result "L is more efficient (for a restricted class of tasks) than L'". Finally, it should be emphasized that a pure expressivity analysis should be seen only as a first step on which investigations of complexity and learnability then can be built.

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

2. An Expressivity Framework

PL-languages define probabilistic models, i.e. probability distributions over some state space. As a first unifying convention, we assume that the state space is generated by a set of random variables that can be written in the syntactic form of ground atoms, e.g. *blood_pressure(tom), sister(susan,tom), genotype(mother(tom)),*... These random variables take values in finite sets of states that are associated with the relation symbol, e.g. *states*(*genotype*)=$\{AA, Aa, aa\}$. At this point we do not consider continuous variables. We call any assignment of states to the set of all ground atoms constructible over a given vocabulary S of relation, function and constant symbols a *Multi-valued Herbrand interpretation*, denoted *MVHI*(S). A *probabilistic-logical model* is a probability distribution over the multi-valued Herbrand interpretations of a given vocabulary.

If we take the foregoing definitions, and restrict attention to vocabularies without function symbols, then any pl-model can be represented by a discrete Bayesian network. On the other hand, all pl-languages can encode Bayesian networks. This seems to imply that all pl-languages and discrete Bayesian networks are equally expressive. In order to see why the solution to our expressivity problem does not reduce to this simple (and unsatisfactory) answer, we have to look more closely at how pl-languages represent pl-models: the power and usefulness of these languages derives from the fact that they encode pl-models in a modular fashion, which separates the specification of a particular, (non-probabilistic) domain structure from the specification of the actual probabilistic relationships. This separation is most clearly expressed in PRMs, where the specification of the *skeleton structure* is distinguished from the actual probabilistic model, and in RBNs, where the specification of an *input structure* is distinguished from the specification of the actual RBN model. Basically the same distinction is represented by the partitioning of a BLP into its *extensional* and *intensional* part.

Adopting database terminology, we base our analysis on the view that models represented in an pl-language can be decomposed into an extensional and an intensional part, where the intensional part contains the generic, high-level probabilistic model, and the extensional part contains specific (typically non-probabilistic) domain information. When translating one pl-language into another, we will require that the translation preserves the modularity of the representation. Figure 1 illustrates the situation: a model M in a language L consists of a pair (M_{int}, M_{ext}) representing the intensional and extensional parts of the representation. It defines a distribution $P = P(M_{int}, M_{ext})$ on MVHIs. To represent M in another language L' as a model (M'_{int}, M'_{ext}), we have to find two separate translations t_{int}, t_{ext} for the intensional and extensional parts.

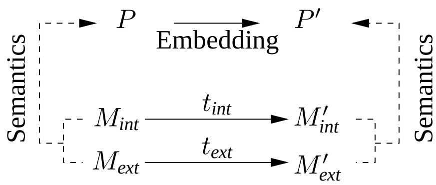

Figure 1. Translations and Embeddings

The translations have to preserve the semantics of the model M. It will usually not be the case that M' represents a distribution over exactly the same multi-valued Herbrand interpretations as M, because the translation will often involve a change of vocabulary (e.g. from a non-binary to a binary encoding). Thus, we cannot require that the distribution P defined by M is equal to the distribution P' defined by M'. All that we can (and should) require is that P can be embedded in P' in the sense of the following definition.

Definition 2.1 *Let P, P' be probability distributions over* MVHI*(S), respectively* MVHI*(S'). An embedding of P in P' is a mapping $h : \mathrm{MVHI}(S) \mapsto 2^{\mathrm{MVHI}(S')}$ such that for each $w \in \mathrm{MVHI}(S)$: $P(w) = P'(h(w))$.*

When P is embedded in P' (written $P \preceq P'$) then every probabilistic query one can pose for P can be answered by computing the solution to a transformed query for P'. The existence of an embedding in the sense of Definition 2.1 is a very strong condition, and sometimes we also need to consider weakened forms: the equality $P(w) = P'(h(w))$ may not be required pointwise for every $w \in \mathit{MVHI}(S)$, but only for a restricted class of subsets $W \subseteq \mathit{MVHI}(S)$. Such a weakened condition then represents the fact that model M and M' are only equivalent with respect to a restricted class of probabilistic queries.

Putting things together, we obtain a formal definition for a partial expressivity order.

Definition 2.2 *Language L' is at least as expressive as L, $L \preceq L'$, if $\exists t_{\text{int}} \forall M_{\text{int}} \exists t_{\text{ext}} \forall M_{\text{ext}}$*

$$P(M_{\text{int}}, M_{\text{ext}}) \preceq P(t_{\text{int}}(M_{\text{int}}), t_{\text{ext}}(M_{\text{ext}}))$$

This definition refines the situation depicted in Figure 1 by allowing the translation t_{ext} to depend on the intensional part of the current model. This asymmetric treatment of intensional and extensional translations reflects the fact that the intensional part is the core of the model, and that extensional parts are just an "add-on" to an intensional specification.

3. First Results and Future Work

It has been shown how MLNs can be embedded in RBNs [4]. This result can be expressed in our framework as $MLN \preceq RBN$. Building on previous results, we next plan to investigate the exact relationships of (functor-free versions) of Prism, RBNs, PRMs, BLPs and MLNs in our framework. This always requires to first identify the extensional and intensional parts in the various representations. For some languages this is straightforward and un-ambiguous. For other languages (e.g. Prism) this is not quite as easy. Another issue that has to be addressed is the exact correspondence between *combination*, as used e.g. in RBNs and BLPs, and *aggregation* as used in PRMs.

The research was supported by the EU IST programme: FP6-508861, Application of Probabilistic ILP II

References

[1] N. Friedman, Lise Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proceedings of IJCAI-99*, 1999.

[2] L. Getoor and J. Grant. PRL: A probabilistic relational language. *Machine Learning*, 62(1-2):7–31, 2006.

[3] M. Jaeger. Relational bayesian networks. In *Proceedings UAI-97*, pages 266–273, 1997.

[4] M. Jaeger. Importance sampling on relational bayesian networks. In *Dagstuhl Seminar Proceeding 05051*, <http://drops.dagstuhl.de/opus/volltexte/2006/411>, 2006.

[5] K. Kersting and L. De Raedt. Towards combining inductive logic programming and bayesian networks. In *Proceedings of ILP-2001*, Springer Lecture Notes in AI 2157, 2001.

[6] S. Muggleton. Stochastic logic programs. In L. De Raedt, editor, *Advances in Inductive Logic Programming*, pages 254–264. IOS Press, 1996.

[7] A. Puech and S.H. Muggleton. A comparison of stochastic logic programs and Bayesian logic programs. In *IJCAI03 Workshop on Learning Statistical Models from Relational Data*. IJCAI, 2003.

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

[9] T. Sato. A statistical learning method for logic programs with distribution semantics. In *Proceedings of ICLP'95*, pages 715–729, 1995.

[10] J. Vennekens and S. Verbaeten. A general view on probabilistic logic programming. In *Proceedings of BNAIC-03*, 2003.