Balios - The Engine for Bayesian Logic Programs

Kristian Kersting and Uwe Dick

University of Freiburg, Institute for Computer Science, Machine Learning Lab,
Georges-Koehler-Allee 079, 79110 Freiburg, Germany
{kersting,dick}@informatik.uni-freiburg.de

1 Context: Stochastic Relational Learning

Inductive Logic Programming (ILP) [4] combines techniques from machine learning with the representation of logic programming. It aims at inducing logical clauses, i.e, general rules from specific observations and background knowledge. Because of focusing on *logical* clauses, traditional ILP systems do not model uncertainty explicitly. On the other hand, state-of-the-art probabilistic models such as *Bayesian networks* (BN) [5], *hidden Markov models*, and *stochastic context-free grammars* have a rigid structure and therefore have problems representing a variable number of objects and relations among these objects. Recently, various relational extensions of traditional probabilistic models have been proposed, see [1] for an overview. The newly emerging field of *stochastic relational learning* (SRL) studies learning such rich probabilistic models.

2 The Balios Engine

Balios is an inference engine for Bayesian logic programs (BLPs) [3, 2]. BLPs combine BNs with definite clause logic. The basic idea is to view logical atoms as sets of random variables which are similar to each other. Consider the modelling the inheritance of a single gene that determines a person's `P` blood type `bt(P)`. Each person `P` has two copies of the chromosome containing this gene, one, `mc(M)`, inherited from her mother `mother(M,P)`, and one, `pc(F)`, inherited from her father `father(F,P)`. Such a general influence relation cannot be captured within BNs.

Knowledge Representation: Like BNs, BLPs separate the qualitative, i.e., the influence relations among random variables, from the quantitative aspects of the world, i.e., the strength of influences. In contrast to BNs, however, they allow to capture general probabilistic regularities. Consider the BLP shown in Figure 1 modelling our genetic domain. The rule graph gives an overview of all interactions (boxes) among abstract random variables (ovals). For instance, the maternal information `mc/1` is specified in terms of mothers `mother/2`, maternal `mc/1` and paternal `pc/1` information. Each interaction gives rise to a local probabilistic model which is composed of a qualitative and a quantitative part. For instance, rule 2 in Figure 1(a) encodes that

> *"the maternal genetic information* `mc(P)` *of a person* `P` *is influenced by the maternal* `mc(M)` *and paternal* `pc(M)` *genetic information of* `P`*'s mother* `M`.

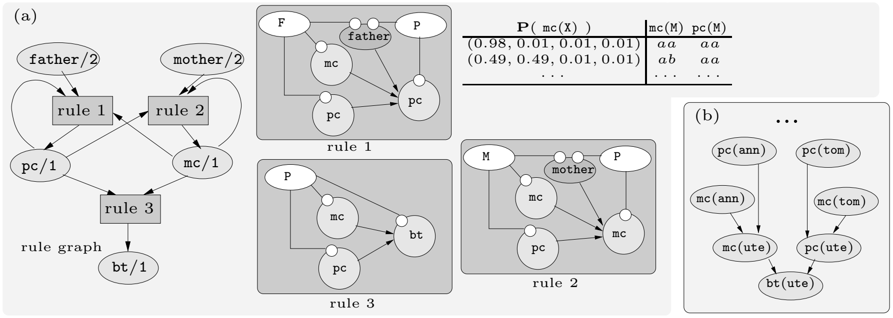

Fig. 1. (a) A graphical BLP. We left out some specification of quantitative knowledge. (b) Parts of the inferred BN specifying the distribution over `bt(ute)`.

Light gray ovals represent abstract random variables such as maternal chromosomes `mc(P)`. Smaller white circles on boundaries denote arguments, e.g., some person `P`. Larger white ovals together with undirected edges indicate that arguments refer to the same person as for `mc(P)` and `mother(M,P)`. To quantify the structural knowledge, *conditional probability distributions* (CPDs) are associated. Some information might be of qualitative nature only, such as `mother(M,P)`. The mother `M` of a person `P` does not affect the CPD but ensures the variable bindings among `mc(P)`,`mc(M)`, and `pc(M)`. Such "logical" nodes are shaded dark gray.

Next to relational probabilistic models, the range of knowledge representation paradigms provided by Balios include e.g. BNs (with purely discrete variables, purely continuous variables, or a mix of discrete and continuous variables), hidden Markov models, stochastic grammars, and logic programs.

Inference: To compute the distribution of a finite set of random variables given some evidence, a BN is inferred. To do so, proof techniques from logic programming are employed because the qualitative structure of a BLP corresponds to a logic program. For instance rule 1 in Figure 1(a) corresponds to the clause `pc(P) :- father(F,P),pc(F),mc(F)` . We assume range-restriction, i.e., each variable `P` in the head `pc(P)` also occurs in the body `father(F,P),pc(F),mc(F)`.

To compute the distribution over `bt(ute)`, we first compute `ute`'s paternal `pc(ute)` and maternal `mc(ute)` information due to rule 3. The associated CPD quantifies the influence. Then, in the next iteration, we deduce the influence relations of the chromosomal information of `ute`'s mother (rule 1) and father (rule 2). In this way, we iterate. This yields a BN, see Figure 1(b), if the influence relation is acyclic. In the presence of evidence, e.g., we know that the blood type of `ute`'s sister `nadine` is `a`, we compute the union of the BNs for `bt(ute)` and `bt(nadine)`, and set `a` as evidence for `bt(nadine)` in the resulting network.

Combining Rules and Aggregate Functions: When there are multiple rules firing, a combining rule such as *noisy-or* or *noisy-and* is used to quantify the combined influence. This also allows for aggregate functions such as *median* and *mode*. Consider modelling the ranking of a new book, see Figure 2. The overall ranking depends on the rankings of individual customers who read the

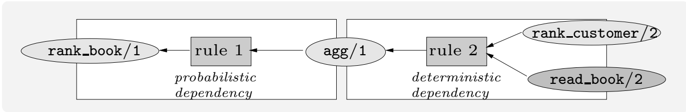

Fig. 2. Modelling the ranking of a book in terms of aggregated individual rankings of customers who read the book.

book. The individual rankings are summarized in `agg/1` which deterministically computes the aggregate property over all customers who read the book. The overall ranking of the book `rank_book/1` probabilistically depends on `agg/1`.

The Engine: BALIOS is written in JAVA. It calls SICSTUS Prolog to perform logical inference and a BN inference engine (e.g. HUGIN or ELVIRA) to perform probabilistic inference. BALIOS features (1) a GUI graphically representing BLPs, (2) compution the most likely configuration, (3) exact (junction tree) and approximative inference methods (rejection, likelihood and Gibbs sampling), and (4) parameter estimation methods (hard EM, EM and conjugate gradient). To the best of the authors' knowledge, BALIOS is the first engine of a turing-complete probabilistic programming language featuring a graphical representation.

3 Demonstration and Concluding Remarks

The demonstration will include examples of Bayesian networks, hidden Markov models, stochastic grammars, and Bayesian logic programs. We will explain the graphical representation, show how to do inference (exact and approximative), and will demonstrate parameter estimation from a database of cases.

At the moment the set of planned future features of BALIOS includes, but is not limited to: effective methods for learning the structure of BLPs, see e.g. [3]; and relational influence diagrams to support decision making.

Acknowledgments: The research was supported by the EU under contract number FP6-508861, *Application of Probabilistic Inductive Logic Programming II*.

References

1. L. De Raedt and K. Kersting. Probabilistic Logic Learning. *ACM-SIGKDD Explorations*, 5(1):31–48, 2003.
2. K. Kersting and L. De Raedt. Adaptive Bayesian Logic Programs. In C. Rouveirol and M. Sebag, editors, *Proceedings of the Eleventh Conference on Inductive Logic Programming (ILP-01)*, volume 2157 of *LNCS*, Strasbourg, France, 2001. Springer.
3. K. Kersting and L. De Raedt. Towards Combining Inductive Logic Programming and Bayesian Networks. In *Proceedings of the Eleventh Conference on Inductive Logic Programming (ILP-2001)*, volume 2157 of *LNCS*. Springer, 2001.
4. S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19(20):629–679, 1994.
5. J. Pearl. *Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 2. edition, 1991.