# An Exercise with Statistical Relational Learning Systems

**Maurice Bruynooghe, Broes De Cat, Jochen Drijkoningen, Daan Fierens, Jan Goos, Bernd Gutmann, Angelika Kimmig, Wouter Labeeuw, Steven Langenaken, Niels Landwehr, Wannes Meert, Ewoud Nuyts, Robin Pellegrims, Roel Rymenants, Stefan Segers, Ingo Thon, Jelle Van Eyck, Guy Van den Broeck, Tine Vangansewinkel, Lucie Van Hove, Joost Vennekens, Timmy Weytjens, Luc De Raedt** {Firstname}.{Lastname}@cs.kuleuven.be
Katholieke Universiteit Leuven, CS Department, Celestijnenlaan 200A, 3001 Heverlee, Belgium

## Abstract

We report on two exercises in modeling, inference and learning with seven statistical relational learning systems and use this as a basis for a simple and preliminary comparison between these systems.

## 1. Introduction

Since the advent of statistical relational learning, a large number of different models, systems and approaches has become available (Getoor & Taskar, 2007; De Raedt et al., 2008). While, on the one hand, this is a sign of the maturity and success of the field, it has, on the other hand, led to difficulties about their differences, similarities and relative strengths, which does seem to hinder progress. We believe, that is the right time to carry out comparisons between existing approaches, which, in turn, should lead to more insight into the general principles underlying the field. The perspective of the comparison taken here is a very pragmatic one. It is centered around state-of-the-art implementations of SRL systems (available in 09/2008). The idea being that rather than providing yet another theoretical result or a comparative evaluation on a specific learning task the goal was to gather concrete experiences with these systems on some simple but realistic tasks. This should provide insight into the usability and limitations of these systems. This idea is not new – it was formulated at some Dagstuhl workshops – and taken up already by Manfred Jaeger and his collaborators, cf. http://www.cs.aau.dk/~jaeger/plsystems/. What is new, however, is that the tasks are more complex and that more systems were considered, such as CLP($\mathcal{BN}$), IBAL and BLOG.

Preliminary work. Do not distribute.

## 2. Experimental Set-Up

The following systems were used

- PRISM (Sato & Kameya, 1997), an extension of Prolog with generative modeling and learning abilities,
- Balios, an implementation of Bayesian Logic Programs (BLPs) (Kersting & De Raedt, 2007) combining Bayesian networks and Prolog using knowledge based model construction,
- the YAP-Prolog implementation of CLP($\mathcal{BN}$) (Santos Costa et al., 2003), combining constraint logic programming with Bayesian networks,
- Alchemy, an implementation of Markov Logic (Richardson & Domingos, 2006) combining first order logic with Markov networks,
- IBAL (Pfeffer, 2007), combining the functional language OCaml with Bayesian networks,
- BLOG (Milch et al., 2007), integrating Bayesian networks with reasoning over unknown objects,
- Primula, an implementation of Relational Bayesian Networks (RBNs) (Jaeger, 1997), combining a relational representation with Bayesian networks.

The evaluation was set up as a graduate course, in which teams of two students carried out two assignments with one of these systems and acted as the "representatives" of their system.

### 2.1. Assignment 1 – an ER-model

In this entity-relationship task, inspired by the PRM tradition (Getoor & Taskar, 2007), the task was to model a real-estate market. There are three types of entities: houses (with attributes such as neighborhood, cost), facilities (like a garden, swimming pool, ... with attributes such as exclusiveness) and customers (with attributes age and rich). The following relationships should be modeled as well: wants (indicates that a customer wants a facility), has (indicates that a house has a facility), interestedin (indicates that a customer is interested in a house) and buy (indicates that a customer buys a house).

In addition to the traditional prior and conditional probabilistic dependencies, the model should also express that 1) the probability that a house is cheap is $p_1^n$ with $n$ the number of facilities of the house; 2) if there is at least one facility that a customer wants, the probability he is interested in a house is $p_2$, otherwise $p_3$; 3) each house is bought by at most one customer.

### 2.2. Assignment 2 – Markov models

For the second assignment, we adopted the HMM example by Russell and Norvig, where a prisoner in an underground jail tries to find out – every day – whether it rains or not. The only observation the prisoner has available is whether his guard is wearing an umbrella. This situation can be modeled as a Hidden Markov Model (HMM) with two hidden states rain (can be true or false) and observations umbrella (again, true or false). The assignment contained also some variants, including a relational one where the prisoner observes an unknown number of guards that might carry an umbrella, and another one requiring the use of a combination or aggregation function.

## 3. Knowledge Representation

The differences in representation power between the systems can – to a large extent – be explained using the following dimensions:

Can the system directly – without auxiliary constructs – represent D1) unconditional probabilities ? D2) conditional probabilities, D3) dynamic probabilities (where the probability is computed dynamically), D4) logical definitions of predicates or relations, D5) continuous random variables, D6) probability distributions over sequences of arbitrary length (as needed to model HMMs), D7) constraints (like Assignment 1, 3) D8) a modular representation of independent causation –when the causes are disjoint, that is, when at most one cause is present (as an example, consider a student failing an exam, this can be because of bad luck with the questions but also because she did not study), D9) when the causes overlap, D10) does the model employ combining or aggregation rules, D11) does the system support structured data types, D12) does it support programming?

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IBAL | Y | Y | N | N | N | Y | N | N | N | N | Y | Y |
| BLOG | Y | Y | Y | Y | Y | Y | N | N | N | N | N | N |
| Primula | Y | Y | N | Y | N | Y | N | Y | Y | Y | N | N |
| PRISM | Y | N | N | Y | N | Y | N | Y | N | N | Y | Y |
| CLP($\mathcal{BN}$) | Y | Y | Y | Y | N | Y | N | N | N | Y | Y | Y |
| Balios | Y | Y | N | Y | Y | Y | N | Y | Y | Y | Y | Y |
| Alchemy | N | N | N | Y | Y | N | Y | Y | Y | N | N | N |

*Table 1.* Knowledge Representation Dimensions

The answers to these questions are indicated in Table 1. While most of the issues are quite clear, it is worth looking into some of the more interesting findings. The first issue is concerned with the representation of (un)conditional probabilities (D1-2), where MLNs do not directly employ probabilities but weights. In this way the logical statements are viewed as "soft constraints" rather than as encoding a generative model, what most of the other systems do. On the one hand, this makes it harder to encode say the ER-model exercise because one has to convert the weights into probabilities, which is in principle possible but in practice very hard. One reason why this is hard is that the probabilities depend on the size of the domain as is the case when modeling the Markov model exercise. In a Markov model the transition probabilities remain the same and do not depend on the length of the sequence, but when representing this with a Markov network (or an MLN, for that matter) the weights depend on the sequence length. The use of the statements as constraints in MLNs has another side-effect: MLNs are the only system that seem able to capture hard constraints (D7, such as that in Assignment 1 – 3). A second issue deserving some attention are the questions surrounding data structures and programming. Being able to encode data structures and to program comes in quite handy to deal with some of the problems. It turns out that this ability has a direct effect on the ability to represent some problems such as dealing with a (Hidden) Markov model over unbounded sequences. Those systems that are embedded in existing programming languages (such as Prolog or OCaml for IBAL) deal with this type of problem directly, whereas the other ones (Primula and Alchemy) exhibit more problems with this. Systems like BLOG can – in principle – deal with structured data at the expense of writing code in the underlying programming language (Java). A third issue is related to the domains of the random variables. BLOG is the only system that deals in elegant manner with continuous domains, although also MLNs and Balios can handle such variables.

## 4. Inference and Learning

In statistical relational learning, not only representation matters, but also inference and learning. Therefore, we also verified to what extent the models could be used for this task.

To learn the parameters of the models, we generated data from a "base model" for the two assignments. In both cases, we employed two different settings. One that corresponded to the fully observable case, the other to the partially observable one. For the ER exercise we provided 100 training and 100 test examples each having 4 houses, 3 customers, and 5 types of facilities. For the Markov model exercise we provided both 10 sequences as training and test set, each having length 10. A subset was used for training in case the implementation could not handle the full datasets.

The way that the systems perform inference and realize learning is quite different. Some dimensions for this include: D13) Is inference based on sampling, D14) on exact computation, or D15) using MPE inference, and for learning, D16) does it support parameter learning, D17) does it use EM, D18) gradient based approach, D19) can it learn form entailment (i.e., use facts as examples), D20) can it learn from interpretations (i.e., possible worlds, relational state-descriptions), D21) does it perform inference in reasonable time for A1, D22) for A2, D23) does it learn in reasonable time for A1, D24) for A2.

| | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IBAL | Y | Y | N | Y | Y | N | Y | N | N | N | Y | Y |
| BLOG | Y | Y | N | N | – | – | – | – | N | N | N | N |
| Primula | Y | Y | N | Y | N | Y | N | Y | Y | Y | N | N |
| PRISM | Y | Y | Y | Y | Y | N | Y | N | N | Y | N | Y |
| CLP($\mathcal{BN}$) | N | Y | N | Y | Y | N | N | Y | Y | Y | N | Y |
| Balios | Y | Y | N | Y | Y | Y | N | Y | Y | Y | N | Y |
| Alchemy | N | N* | Y | Y | Y | Y | N | Y | Y | N | N | N |

*Table 2.* Different dimensions for Inference and Learning

One conclusion of our investigation is, that although the exercises were simple, it is often not easy to model them with the available SRL languages. And if one succeeds in the modeling task, both inference and definitely learning may still encounter problems. When we started our investigation we actually intended to go beyond these simple exercises and did not really expect to encounter so many difficulties, because, after all, SRL is now almost 10 years old now. While on the one hand, this is surprising, it also in a sense explains why there are so many different approaches in SRL. What is easy with one system, is often harder with another system. This situation is akin to that with programming and knowledge representation languages. On the other hand, this state-of-affairs also shows that the field is in need of proper frameworks, evaluation measures and benchmarks that allow to compare SRL systems along different dimensions, like expressivity, modeling power, efficiency of inference and learning

## 5. Conclusions

With SRL becoming more mature, we thought it was the right time to carry out a simple evaluation of state-the-art SRL systems using two exercises. The results provided some useful insights into SRL.

## Acknowledgments

We thank the developers of the systems for their help. This work was supported by the GOA/08/008 "Probabilistic Logic Learning", by the Research Foundation-Flanders (FWO-Vlaanderen) and by IWT Vlaanderen.

## References

De Raedt, L., Frasconi, P., Kersting, K., & Muggleton, S. (Eds.). (2008). *Probabilistic inductive logic programming — theory and applications*. Springer.

Getoor, L., & Taskar, B. (Eds.). (2007). *An introduction to statistical relational learning*. MIT Press.

Jaeger, M. (1997). Relational Bayesian networks. *UAI* (pp. 266–273). Morgan Kaufmann.

Kersting, K., & De Raedt, L. (2007). Bayesian logic programming: theory and tool. In (Getoor & Taskar, 2007).

Milch, B., Marthi, B., Russell, S., Sontag, D., Ong, D. L., & Kolobov, A. (2007). BLOG: Probabilistic models with unknown objects. In (Getoor & Taskar, 2007).

Pfeffer, A. (2007). The design and implementation of IBAL: A general-purpose probabilistic language. In (Getoor & Taskar, 2007).

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning*, *62*, 107–136.

Santos Costa, V., Page, D., Qazi, M., & Cussens, J. (2003). CLP($\mathcal{BN}$): Constraint logic programming for probabilistic knowledge. *UAI* (pp. 517–524).

Sato, T., & Kameya, Y. (1997). PRISM: A symbolic-statistical modeling language. *IJCAI* (pp. 1330–1339). Morgan Kaufmann.