JudgeD: a Probabilistic Datalog with Dependencies

A presentation at the Dutch-Belgian Database Day 2015.

Speaker: Brend Wanders <b.wanders@utwente.nl>
Affiliation: Database Group, Faculty of EEMCS, University of Twente

Abstract

Over the past three decades several probabilistic logics have been developed. Prominent examples are Probabilistic Horn Abduction, Stochastic Logic Networks, and two probabilistic datalogs known as pD and ProbLog. In these logics probabilities can be attached to logical formulas, under the imposition of various constraints. During the same period, several relational probabilistic databases have been developed such as MayBMS, Trio and MCDB. As with the probabilistic logics, certain constraints are imposed.

We present JudgeD, a probabilistic datalog in which probabilities can be attached to both factual data and rules. The key contributions of JudgeD are: the expression of dependencies between arbitrary clauses, both facts and rules (e.g., mutual exclusivity, independence, mutual dependence, implication and more complex dependency relations), and secondly the proof-of-concept implementation of both a Monte Carlo based approximation as well as an exact solver.

A motivating example for the development of JudgeD is its use as reasoning system for the combination of uncertain evidence about maritime data. The case described in [1] has as ultimate goal the automatic determination of the chance that an observed vessel is engaged in smuggling based on a observations about these vessels.

The formal underpinning of JudgeD is a data-model independent framework of probabilistic databases. In the presentation we will give an intuitive overview of the formalisation, a more in-depth discussion of the formal underpinnings of JudgeD can be found in [2].

The proof-of-concept implementation of JudgeD is based on SLG resolution for negative Prolog. The focus of the implementation is not on raw performance, but on ease of prototyping, as such the system is implemented in the Python language to allow for quick prototyping of new approaches. The proof-of-concept implementation offers two methods of evaluation: a Monte Carlo approximation, and an exact solver.

References

[1] M. B. Habib, B. Wanders, J. Flokstra, and M. van Keulen. Data uncertainty handling using evidence combination: a case study on maritime data reasoning. In *Proceedings of the 5th DEXA Workshop on Information Systems for Situation Awareness and Situation Management (ISSASiM 2015), Valencia, Spain*. IEEE Computer Society, September 2015.

[2] B. Wanders and M. van Keulen. Revisiting the formal foundation of probabilistic databases. In *Proceedings of the 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology*. Atlantis Press, June 2015.