CLP($\mathcal{BN}$) In School: Inductive Logic Programming and Bayesian Networks

Vítor Santos Costa, David Page
Biostatistics and Medical Informatics
University of Wisconsin-Madison

James Cussens
Department of Computer Science
University of York

We present CLP($\mathcal{BN}$), a novel approach that aims at integrating Bayesian variables with the logic programming framework. Our motivation in developing CLP($\mathcal{BN}$) was to reason with relations between probabilistic and non-probabilistic knowledge.

Logic programming traditionally uses constraints to represent partial knowledge about variables. CLP($\mathcal{BN}$) is based on two key observations. First, we observe that random variables are bound variables. The only difference is is that we do not know to which value they were bound: instead, we know that the value is an (unknown) function of the universally quantified variables in the clause. We thus constrain a random variable to be a skolem function of the universally bound variables. Our second observation is that even if we do not know the actual value of the random variable, we often have a probability distribution for the possible values of the skolem function. A second constraint represents such a probability distribution. Succesful execution of a CLP($\mathcal{BN}$) program returns a store consisting of a network of skolem functions related through probability distributions. This store thus forms a Bayesian network, and parameters of interest can be obtained through standard Bayesian solving techniques.

We have successfuly experimented on CLP($\mathcal{BN}$) with examples such as an artificially generated school data-based, in the style of the Probabilistic Relational Models (PRM) example.

In a second step we have experimented with learning CLP($\mathcal{BN}$) programs in the presence of complete data. We use likelihood as our basic measure. We extend Srinivasan's Inductive Logic Programming System Aleph to learn CLP($\mathcal{BN}$) programs. Our initial experiment consists of learning the relations that define random variables in the school data-base. We use a greedy algorithm to do multi-predicate learning. Initial results show that Aleph can indeed replicate most of the rules we used to generate the original data-base.