Graphical models, an articulation point between Constraint Programming & Machine Learning

Position paper by Thomas Schiex

Graphical models (GMs[1]) represent a family of mathematical models that allow to concisely represent multivariate probability distributions based on specific conditional independence assumptions represented in a (possibly directed) graph or hypergraph. In these models, the joint probability distribution is represented as the product of local functions, often called factors or potential functions, each involving few variables (often less than 2). Reminds you of something?

Probabilistic graphical models cover essentially two families of models: Bayesian networks using a directed graph and Markov random fields using an undirected graph (also chain/factor/ancestral graphs, conditional random fields, restricted Boltzmann machine and influence diagrams). With continuous variables using Gaussian distributions, they are called Gaussian Graphical Models. Restricted Boltzmann machines have also been used for "Deep Learning".

Several computational problems in discrete GM processing can exploit a known GM to answer queries such as: what is the total (MAP/MPE) or partial (Marginal MAP) assignment with maximum probability or what is the probability that a subset of all variables takes some value (MAR)... These problems all reduce to NP-complete discrete optimization and #P-complete counting or a combination of these. Because of these complexities, most algorithms are heuristics (eg. the famous Loopy Belief Propagation algorithm, which does not always converge).

Graphical models can be learnt: it is possible to estimate the potential functions and even the graphical structure (the scopes of the functions) from data, paving the way to (big) data analytics.

Being widely used in machine learning, image processing, statistical physics and statistics... GM processing papers appear at Uncertainty in Artificial Intelligence (UAI), Neural Information Processing (NIPS), Computer Vision and Pattern Recognition (CVPR) as well as in IEEE Pattern Analysis and Machine Intelligence journal (one of the strongest impact factor in AI).

Similarly, a constraint satisfaction problem is a mathematical model that allows to concisely represent a boolean "feasibility" distribution as a product (logical and) of local boolean functions (constraints). They are Markov random fields with just $\{0, 1\}$ potential functions. They are graphical models. Similarly, weighted constraints (or cost function) networks are essentially non-negative integer or rational valued $-\log$ Markov random fields.

Here, satisfaction, (quantified) optimization and counting are the main problems with an emphasis on well characterized algorithms (such as local consistency enforcing, including global constraints or cost functions) and data-structures (decision diagrams), as well as algorithms and data-structures giving access to exact or guaranteed solutions.

Starting in UAI 2010, and then 2012, 2014, exact Weighted CSP solvers such as `toulbar2` and `daoopt` have been systematically winning the *Approximate* Probabilistic Inference Challenge in the optimization (MPE/MAP) category and sometimes even in the marginal optimization category (Marginal MAP, mixing counting and optimization). Some of these benchmark instances have half a million non boolean variables and yet, can be sometimes solved to optimality.

The ability of GMs to represent uncertainty, to be learnt from/fitted to data but also as a support for decision making as in CP make them ideal modeling and processing tools in the chain that starts from data to decision. The main challenges in implementing this chain are knowledge representation (there are only few graphical model programming languages) and computational complexity, two challenges that CP has been addressing since its very beginning.

[1] Wikipedia link.