Abstract: Success Stories of CP in Data Mining

Tias Guns
July 18, 2016

The use of constraint programming technology in data mining is increasing. Especially in constraint-based pattern mining, where the goal is to find certain patterns in data, and constraint-based clustering, where the goal is to find a partition of the data.

The main trade-off in these lines of research is efficiency versus generality. Interestingly, in data mining, many highly efficient search methods are known, and a large body of research studies how to make them more generic. Oppositely, in constraint programming many generic search methods are known and a large body of research studies how to increase efficiency.

Just as one can make specific data mining systems more generic through effort and expertise, so can generic constraint solving systems be made as efficient, or more, than specialized mining systems. This is something that few people, especially in data mining, seem to realize.

This is a great opportunity for constraint programming though. The following four success stories aim to exemplify this.

Success 1: constrained itemset mining.
In [4] we used a generic constraint solver to handle more constraint-based mining variants than the state-of-the-art. It was not very scalable, but outperformed specialized methods on complex tasks. In [9] a related branch-and-bound problem was addressed. An advanced bounds-consistent global propagator was introduced and the generic system outperformed the state-of-the-art in many cases. A special purpose algorithm with the same ideas outperformed both. At CP this year [7], a global constraint is proposed for a key component of all of the above, which improves scalability a lot.

Success 2: constrained sequence mining.
In [8], a generic CP approach to find constrained sequences in databases is proposed, but scalability was limited even when using global constraints. Soon after, [6] proposed a single global constraint that was far more scalable, and could outperform the state-of-the-art when considering regular expression constraints. Recently, we improved this global constraint to the point that a generic constraint solver outperforms all existing sequence mining methods [1].

Success 3: constrained clustering.
In [2], CP is used for exact constrained clustering. In contrast to other approaches, it supports all common clustering constraints as well as multiple objective functions. Using global constraints for the objective functions (distance functions) achieves better efficiency then other exact techniques [3]. For the challenging problem solved heuristically by k-means, an iterative CP approach [5] can outperform other exact methods and it supports additional constraints.

Outlook. The above three data mining problems are problems that have constraints by nature, and where one is looking for all (enumeration) or the optimal solution. Both properties fit constraint programming really well.

Discovering and understanding potential benefits of CP for data mining is one issue, but explaining the benefits to data miners is another. One way that has been working well for me recently is to describe CP solvers as highly optimized, modular depth-first search engines. Highly optimized in their handling of backtracking and state, yet modular with the ability to add arbitrary global constraints.

While this is far from the declarative ideals of CP, it is close to current needs in data mining. It can also be a step towards achieving general-purpose approaches for both research fields.
References


