

Constraint Programming for Correlated Itemset Mining

Siegfried Nijssen

Tias Guns

Luc De Raedt

Departement Computerwetenschappen
K.U. Leuven
Celestijnenlaan 200A, 3000 Leuven, Belgium

Abstract

Discovering itemsets and conjunctive rules under constraints are popular topics in the data mining and machine learning communities, for which many algorithms have been proposed. Despite the abundance of research in this area, however, constraint programming (CP) techniques developed in the artificial intelligence community to deal with constraint satisfaction problems have never been applied to rule discovery. In [4], we show that CP can not only be applied in an intuitive, extendible way to rule discovery, but also that CP techniques significantly outperform existing approaches in data mining.

1 Introduction

Both in machine learning and data mining a popular topic of research is learning rule-based classifiers. A rule-based classifier consists of a set of rules of the kind

if income=high and debt=low **then** accept=yes.

An essential step in building such classifiers is to discover rules that predict the target attribute well. A common approach in the machine learning community is to learn one such rule by applying a heuristic in a greedy algorithm. In the data mining community, on the other hand, it has been studied how to find such rules under constraints in an optimal way. The traditional example is the search for all *association rules*. Usually however many association rules can be found and their direct application for classification is impossible. To focus the discovery of rules more towards classification, the use of correlation constraints has been studied, leading to algorithms for *correlated* or *discriminative* itemset mining [3, 1]. Assumed given is a function f which scores every itemset based on how well it correlates with a target attribute. Examples of such functions are χ^2 and information gain. The problem is to find the k itemsets that score highest with respect to f . For $k = 1$, this problem can be thought of as finding the optimal rule under function f , instead of an arbitrary good one, as common in machine learning. This problem is known to be NP-complete, and hence, a general, efficient algorithm cannot be expected to exist.

An area in artificial intelligence which has studied hard constraint satisfaction problem solving extensively, is that of constraint programming [5]. Main principles in constraint programming are:

- problems are specified declaratively by providing constraints on variables within domains;
- solvers find solutions by constraint propagation.

Constraint propagation is the process of reducing domains of some variables based on constraints and domains of other variables; for instance, if $X, Y \in \{0, 1\}$ and $X < Y$, then we can derive that $X = 0$.

Due to their generality, CP systems have been applied successfully in many applications –scheduling in particular. It has however never been applied to rule learning problems; its application in machine learning in general is also rare. Our main contribution is that we show how to apply CP systems in rule learning problems in such a way that the CP system outperforms the state-of-the-art in data mining.

Dataset	CP	[1]	[3]	Dataset	CP	[1]	[3]
anneal	0.22	22.46	24.09	letter	52.66	–	>
australian-credit	0.30	3.40	0.30	mushroom	14.11	0.09	13.48
breast-wisconsin	0.28	96.75	0.28	primary-tumor	0.03	0.26	0.13
diabetes	2.45	–	128.04	segment	1.45	–	>
heart-cleveland	0.19	9.49	2.15	soybean	0.05	0.05	0.07
hypothyroid	0.71	–	10.91	splice-1	30.41	1.86	31.11
ionosphere	1.44	–	>	vehicle	0.85	–	>
kr-vs-kp	0.92	125.60	46.20	yeast	5.67	–	781.63

Table 1: Runtimes, in seconds, of 3 top-1 correlated itemset miners, on an Intel Core 2 Duo E6600 and 4GB of RAM; >: experiments timed out after 900s. –: experiments failing due to memory overflow.

2 Itemset Mining as Constraint Programming

Given a set of items (or binary attributes) \mathcal{I} , a set of transactions (or examples) \mathcal{T} , a transaction database can be considered a subset $\mathcal{D} \subseteq \mathcal{T} \times \mathcal{I}$. A bi-set is a tuple (I, T) with $I \subseteq \mathcal{I}$ and $T \subseteq \mathcal{T}$. An itemset I covers a transaction set T iff $T = \varphi(I)$ where $\varphi(I) = \{t \in \mathcal{T} \mid \forall i \in I : (t, i) \in \mathcal{D}\}$. Hence, an itemset can be seen as a conjunctive rule. A bi-set (I, T) is a *frequent itemset* if it satisfies two constraints: $T = \varphi(I)$ and $|T| \geq \theta$, where θ is a user defined threshold. In [2] we showed that we can formulate these constraints in a constraint program by introducing a variable I_i for each $i \in \mathcal{I}$ and a variable T_t for each $t \in \mathcal{T}$, and expressing the constraints as follows:

$$\forall t \in \mathcal{T} : T_t = 1 \leftrightarrow \sum_{i \in \mathcal{I}} I_i (1 - \mathcal{D}_{ti}) = 0 \quad \text{and} \quad \sum_{t \in \mathcal{T}} T_t \geq \theta.$$

Here $\mathcal{D}_{ti} = 1$ if $(t, i) \in \mathcal{D}$ and $\mathcal{D}_{ti} = 0$ if $(t, i) \notin \mathcal{D}$. It is straightforward to implement these constraints in a CP system, such as Gecode [5]. Using the propagators readily available in Gecode, we can hence search for all itemsets satisfying these constraints. The resulting search can be shown to be similar to that of known itemset mining systems in the data mining community.

To apply CP on correlated itemset mining, where examples belong to two classes \mathcal{T}^+ and \mathcal{T}^- , we need to modify this program. We assume given a function $f(p, n)$ that scores every itemset, based on how many positive and negative examples it covers, and a threshold θ on correlation. The problem of correlated itemset mining is to find all bi-sets for which $T = \varphi(I)$ and $f(|T \cap \mathcal{T}^+|, |T \cap \mathcal{T}^-|) \geq \theta$. In [4] we show that for functions such as information gain and χ^2 , we can also formulate this as:

$$\forall t \in \mathcal{T} : T_t = 1 \leftrightarrow \sum_{i \in \mathcal{I}} I_i (1 - \mathcal{D}_{ti}) = 0 \quad \text{and} \quad \forall i \in \mathcal{I} : I_i = 1 \rightarrow f\left(\sum_{t \in \mathcal{T}^+} T_t \mathcal{D}_{ti}, \sum_{t \in \mathcal{T}^-} T_t \mathcal{D}_{ti}\right) \geq \theta.$$

For the second constraint we need to add a propagator to the CP system, based on an evaluation of the function f in ROC space. We can also adopt this formulation for top-1 itemset mining. Table 1 illustrates how this approach, when implemented in Gecode, compares to existing algorithms in the data mining community.

These experiments show convincingly that the CP approach often outperforms existing algorithms. We showed that itemset mining can be formulated in an extendible way in CP systems. These promising results convince us that the relationships between machine learning, data mining and CP deserve further studies.

References

- [1] H. Cheng, X. Yan, J. Han, and P.S. Yu. Direct discriminative pattern mining for effective classification. In *ICDE*, pages 169–178, 2008.
- [2] L. De Raedt, T. Guns, and S. Nijssen. Constraint programming for itemset mining. In *KDD*, pages 204–212, 2008.
- [3] S. Morishita and J. Sese. Traversing itemset lattice with statistical metric pruning. In *PODS*, pages 226–236, 2000.
- [4] S. Nijssen, T. Guns, and L. De Raedt. Correlated itemset mining in ROC space: A constraint programming approach. In *KDD*, 2009.
- [5] C. Schulte and P.J. Stuckey. Efficient constraint propagation engines. *ACM Trans. Program. Lang. Syst.*, 31(1), 2008.