

Constraint Programming for Data Mining and Machine Learning



Luc De Raedt and *Tias Guns* and Siegfried Nijssen

DTAI, K.U.Leuven, Belgium

AAAI10 nectar, based on papers presented at KDD08 and KDD09

Data Mining and Machine Learning

- Numerous constraints have been defined
- Numerous systems have been developed

Yet,

- new constraints mostly require new implementations
- very hard to combine different constraints

Constraints

- Data Mining & Machine Learning
specific use of constraints in algorithms
- Constraint Programming
general methodology for handling constraints

Surprisingly, CP has not been applied
on Data Mining & Machine Learning

Constraint Programming

One of the succes stories of A.I.

model:

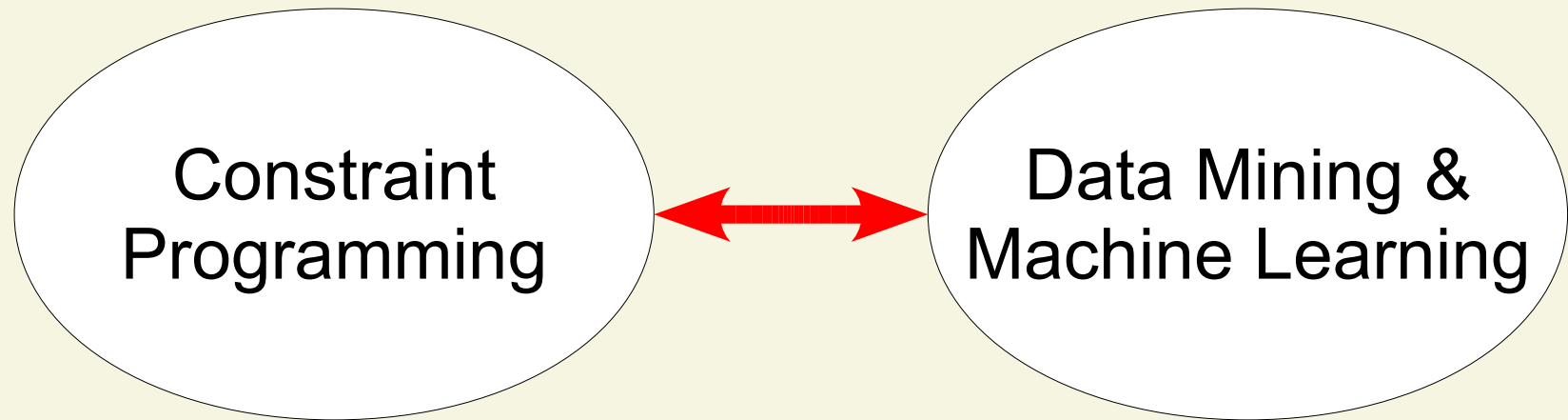
declarative specification of constraints

+

search:

generic handling of variables and constraints
& efficient propagation of individual constraints

Challenge



- Can CP be used for DM & ML?
- Can CP compete in and contribute to DM & ML?

Constraint-based mining

Use of constraints in data mining
to specify the desired set of solutions

(Mannila & Toivonen, 1997)

$$Th(L, p, D) = \{ \phi \in L \mid p(\phi, D) = true \}$$

L Language

p Constraints, *declarative* by nature

D Data

ϕ Pattern

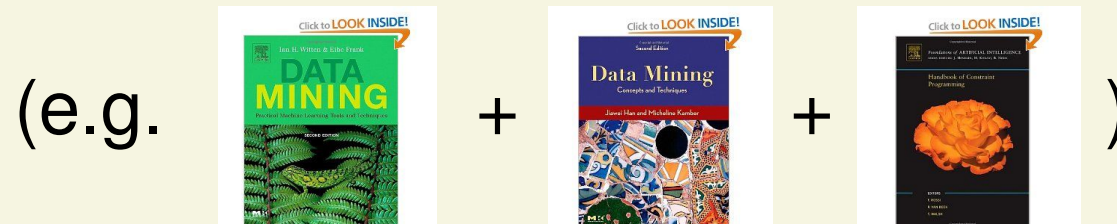
Motivation

Analysing a dataset
to find patterns of interest

For example:

Analysing purchases (e.g. books)

Here, patterns are sets of “items”



Patterns of interest:

- which patterns are frequent ?
- which patterns have a high average price ?
- which patterns are non-redundant ?
- which patterns are frequent on one dataset and infrequent on the other ?
- which patterns correlate with a class label ?
- which patterns are significant w.r.t a background model ?

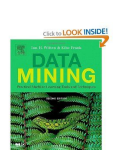
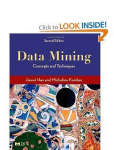
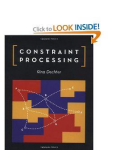
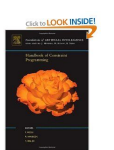



→ specified by constraints

Overview

1. Motivation

2. Frequent itemset mining
3. Constraint-based itemset mining
4. Experiments
5. Conclusions

Frequent Itemset Mining




				
	1	0	1	1
	1	1	0	1
	0	0	1	1

$\text{cover}(\text{Constraint Processing}, \text{Constraint Processing}) = \{ \text{Person with hat}, \text{Person with black hair} \}$

$\text{frequency}(\text{Constraint Processing}, \text{Constraint Processing}) = 2$

CP for Itemset Mining



		i1	i2	i3	i4
		0/1	0/1	0/1	0/1
 t1	0/1	1	0	1	1
 t2	0/1	1	1	0	1
 t3	0/1	0	0	1	1

coverage: $\forall T_t: T_t = 1 \Leftrightarrow \bigwedge_{i, D_{ti}=0} \neg I_i$

frequency: $\forall I_i: I_i = 1 \Rightarrow \sum_{t, D_{ti}=1} T_t \geq \text{Freq}$

CP4IM, basic model

Algorithm 1 Fim_cp's frequent itemset mining model, in **Essence'**

```
1: given NrT, NrI : int
2: given TDB : matrix indexed by [int(1..NrT),int(1..NrI)] of int
3: given Freq : int
4: find Items : matrix indexed by [int(1..NrI)] of bool
5: find Trans : matrix indexed by [int(1..NrT)] of bool
6: such that
7: $ encode TDB: every Trans its complement has no supported Items
8: forall t: int(1..NrT).
9:    $Trans[t] \iff ((\text{sum } i: \text{int}(1..NrI). Items[i]*(1-TDB[t,i])) = 0),$ 
10: $ frequency: every Item is supported by sufficiently many Trans
11: forall i: int(1..NrI).
12:    $Items[i] \implies ((\text{sum } t: \text{int}(1..NrT). Trans[t]*TDB[t,i]) \geq Freq)$ 
```

CP for Itemset Mining

coverage: $\forall T_t: T_t=1 \Leftrightarrow \bigwedge_{i, D_{ti}=0} \neg I_i$

freq ≥ 2 : $\forall I_i: I_i=1 \Rightarrow \sum_{t, D_{ti}=1} T_t \geq \text{Freq}$

- propagate i2

Intuition: infrequent

i2 can never be part of freq. superset

	i1	i2	i3	i4
	0/1	0	0/1	0/1
t1 0/1	1	0	1	1
t2 0/1	1	1	0	1
t3 0/1	0	0	1	1

CP for Itemset Mining

coverage: $\forall T_t: T_t = 1 \Leftrightarrow \bigwedge_{i, D_{ti}=0} \neg I_i$

freq ≥ 2 : $\forall I_i: I_i = 1 \Rightarrow \sum_{t, D_{ti}=1} T_t \geq \text{Freq}$

- propagate i2
- propagate t1

Intuition: unavoidable

t1 will always be covered

		i1	i2	i3	i4
		0/1	0	0/1	0/1
t1	1	1	0	1	1
t2	0/1	1	1	0	1
t3	0/1	0	0	1	1

CP for Itemset Mining

coverage: $\forall T_t: T_t = 1 \Leftrightarrow \bigwedge_{i, D_{ti}=0} \neg I_i$

freq ≥ 2 : $\forall I_i: I_i = 1 \Rightarrow \sum_{t, D_{ti}=1} T_t \geq \text{Freq}$

- propagate i2
- propagate t1
- branch i1=1

		i1	i2	i3	i4
		1	0	0/1	0/1
t1	1	1	0	1	1
t2	0/1	1	1	0	1
t3	0/1	0	0	1	1

CP for Itemset Mining

coverage: $\forall T_t: T_t = 1 \Leftrightarrow \bigwedge_{i, D_{ti}=0} \neg I_i$

freq ≥ 2 : $\forall I_i: I_i = 1 \Rightarrow \sum_{t, D_{ti}=1} T_t \geq \text{Freq}$

- propagate i2
- propagate t1
- branch i1=1
- ...

		i1	i2	i3	i4
		1	0	0/1	0/1
t1	1	1	0	1	1
t2	0/1	1	1	0	1
t3	0	0	0	1	1

More constraints

- Coverage (required) $T_t = 1 \Leftrightarrow \sum_i I_i (1 - D_{ti}) = 0$
- Frequent $I_i = 1 \Rightarrow \sum_t T_t D_{ti} \geq \text{Freq}$
- Maximal $I_i = 1 \Leftrightarrow \sum_t T_t D_{ti} \geq \text{Freq}$
- Closed $I_i = 1 \Leftrightarrow \sum_t T_t (1 - D_{ti}) = 0$
- Delta-closed $I_i = 1 \Leftrightarrow \sum_t T_t (1 - \delta - D_{ti}) = 0$
- ...

+ combinations !

Correlated itemset mining

Also known as: **discriminative itemset mining**, contrast set mining, emerging itemsets, subgroup discovery, ...

- **Given:** *labelled* transactions



- Find: the itemset that *best correlates* with the class label

$$\langle \text{Data Mining} \rangle : \{+, \times\}$$

$$\langle \text{Data Mining}, \text{Data Mining} \rangle : \{++, \times, \}$$

Correlation constraint

$$f\left(\sum_{t \in P} T_t, \sum_{t \in N} T_t\right) \geq \text{Bound}$$

- Existing pruning technique:

only uses upper-bound of $\sum T$

- Our CP-based propagator:

uses upper- and lower-bound of $\sum T$
and look-ahead formulation $I_i = 1 \Rightarrow \dots$

much stronger propagation !

CP4IM software

Based on open-source Gecode library for CP



- C++, very efficient, well documented
- Generic and extensible

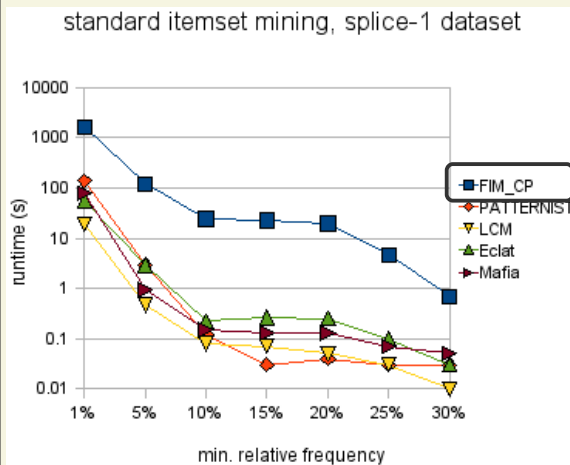
Constraint Programming for Itemset Mining



- Also open-source and extensible
- Many constraints and documentation

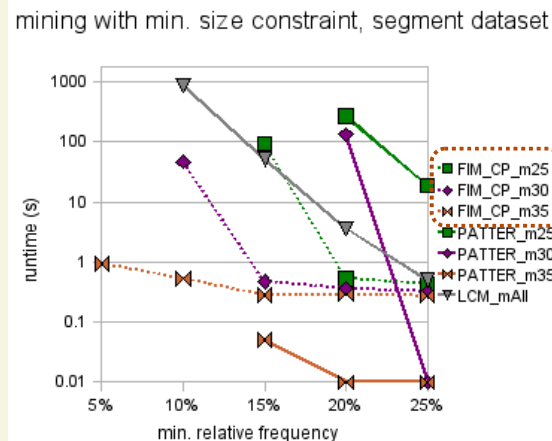
Experiments with CP4IM

Basic setting



Overhead

Strongly constrained



Competitive

Correlated

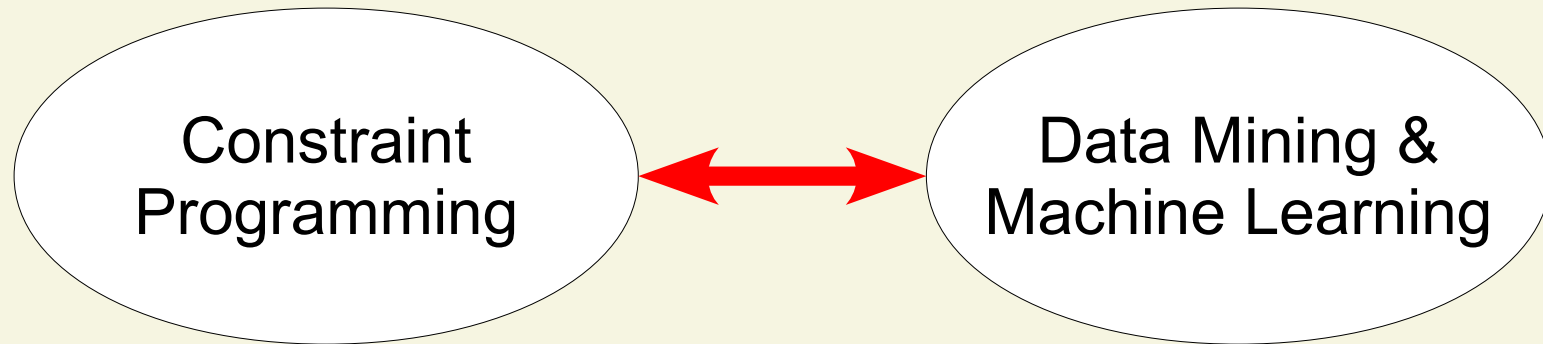
Dataset	CP	(Cheng et al. 2008)	(Morishita and Sese 2000)
anneal	0.22	22.46	24.09
australian-credit	0.30	3.40	0.30
breast-wisconsin	0.28	96.75	0.28
diabetes	2.45	-	128.04
heart-cleveland	0.19	9.49	2.15
hypothyroid	0.71	-	10.91
ionosphere	1.44	-	>
kr-vs-kp	0.92	125.60	46.20
letter	52.66	-	>
mushroom	14.11	0.09	13.48
primary-tumor	0.03	0.26	0.13
segment	1.45	-	>
soybean	0.05	0.05	0.07
splice-1	30.41	1.86	31.11
vehicle	0.85	-	>
yeast	5.67	-	781.63

Best

Generality of CP4IM

	LCM [15]	MAFIA [6]	ExAMiner [4]	DualMiner [5]	DDPmine [12]	CP4IM
Constraints on data						
Minimum frequency	X	X	X	X		X
Maximum frequency				X		X
Emerging patterns						X
Condensed Representations						
Maximal	X	X		X		X
Closed	X	X				X
δ -Closed						X
Constraints on syntax						
Max/Min total cost			X	X		X
Minimum average cost			X			X
Max/Min size	X	X	X	X		X
Constraints on labelled data						
Minimum correlation					X	X
Maximum correlation						X

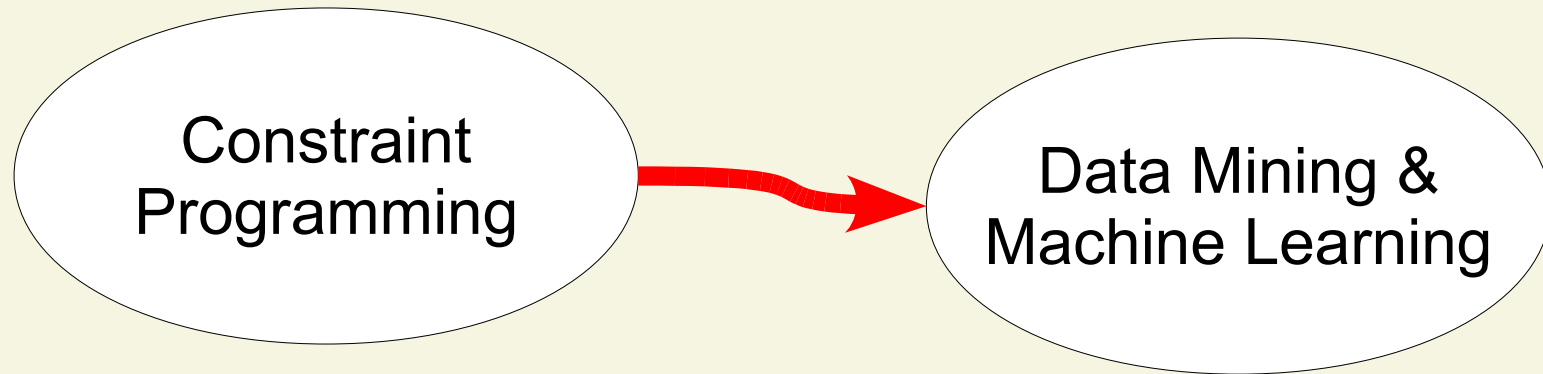
Challenge



- Can CP be used for DM & ML?
- Can CP compete in and contribute to DM & ML?

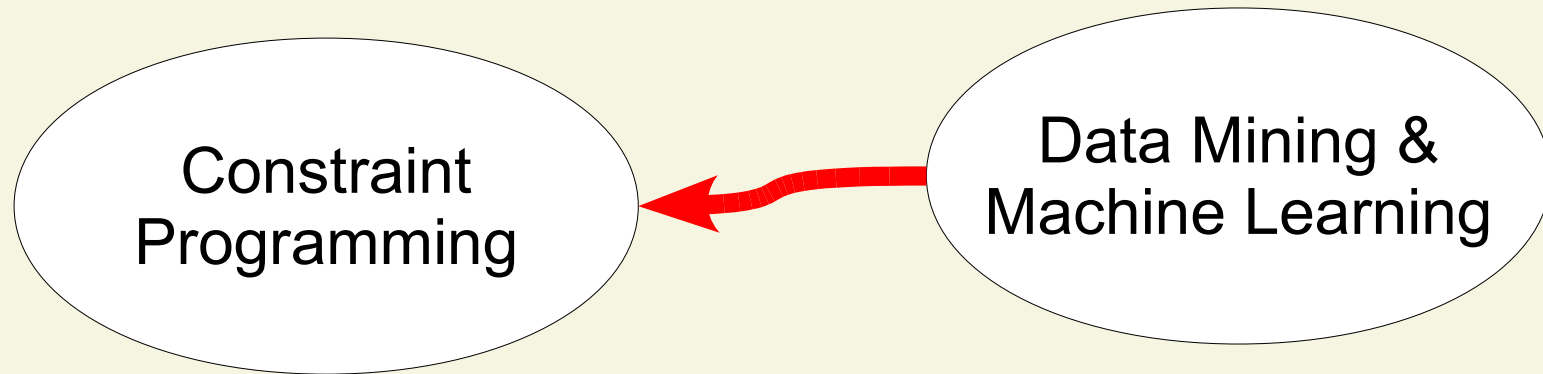
Yes, and itemset mining is just the beginning

What CP offers to DM & ML



- Declarative: model + search
- Flexible: independent propagators
- General: easily combining constraints
- Rapid prototyping, iterative process

What DM & ML offer to CP



- New applications and challenging benchmarks
- Efficient specialized algorithms
- New data and pattern types:
clusters, decision trees, graphs, ...

Thank you for listening

Constraint Programming



Data Mining & Machine Learning

Questions?

<http://dtai.cs.kuleuven.be/CP4IM>

CP4IM Constraint Programming for Itemset Mining

HOME Downloads FM_CP DMCP Datasets Publications

Welcome to CP4IM: Constraint Programming for Itemset Mining

This website aims to gather information about the usage of Constraint Programming in Itemset Mining and Pattern Mining in general. Publications, datasets, software and extra documentation are all available on this website.

Constraint-based itemset mining
Mining all itemsets that satisfy the constraints

FM_CP is the most flexible itemset mining framework to date. The declarative language allows one to express and combine many different constraints. The constraint solver will use these constraints effectively to prune the search space. Some capabilities of the framework:

- **Different interdependencies measures** including frequent itemsets, discriminatory itemsets and emerging itemsets.
- **Different condensed representations** including closed itemset mining (formal concept learning), delta-closed itemset mining and maximal itemset mining.
- **Different item constraints** where one can put a threshold on both the minimum or the maximum value. Examples of properties that can be constrained are the size of the itemset, or in case the cost over every item is known, the total or average cost of the itemset.
- **Combining constraints** all of the above constraints can be easily combined, as well as any other constraint that one can express. The constraint solver guarantees the correct evaluation of the whole.

More information in the documentation and the paper.

Discriminative itemset mining
Mining the top-k itemsets w.r.t. a correlation function

DMCP is the discriminative/correlated itemset mining framework with the most effective pruning to date.

 - **Any convex or monotone function** using the number of positive and negative examples covered can be used. Examples of power functions are information gain, chi-square, gini index and fisher score. Examples of monotone functions are accuracy, relative accuracy and recall.
 - **Many different domains** use this kind of functions, including context-sensitive mining, emerging pattern mining and subgroup discovery, but it has also been named k-optimal pattern discovery, interesting itemset mining, emerging itemset mining, discriminative itemset mining and correlated itemset mining. Lastly, many of the functions used have their origin in rule learning.
 - **Other constraints** can be easily added to the search, as DMCP is built on FM_CP.

More information in the documentation and the paper.

FM_CP:

 - Latest version: FM_CP 2.1
 - Documentation

DMCP:

 - Latest version: DMCP 2.1
 - Documentation

March 2009 43 / 47