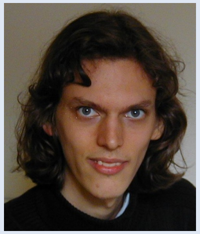
Constraint Programming for Itemset Mining

Tias Guns, K.U.Leuven, Belgium

In collaboration with Siegfried Nijssen *and* Luc De Raedt



Based on papers at KDD08 and KDD09

Position in summer school

Itemset Mining (Bart Goethals' talk)

- Apriori (Level-wise search, anti-monotonicity)
- Eclat (Specific depth-first search)
-

?

Constraint Programming

- Combinatorial Satisfaction Problems (CSP)
- Generic depth-first search

Constraint Programming for Itemset Mining

I. Motivation, constraint-based mining

II. Constraint Programming basics

III. Constraint-based itemset mining using CP

IV. Correlated itemset mining using CP

V. Conclusions.

(frequent) Itemset mining

Transactions:

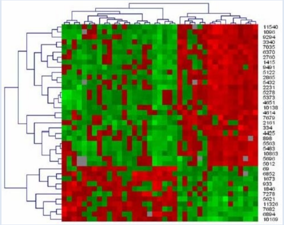
Goal: find patterns in transactional data

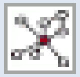
- better understanding of data
- find novel information

Solution: Itemset Mining

Applications:

- online shops
- weblog analysis
- microarray analysis (gene expression)
- learning taxonomies
- text analysis (privacy leaks)
- ...

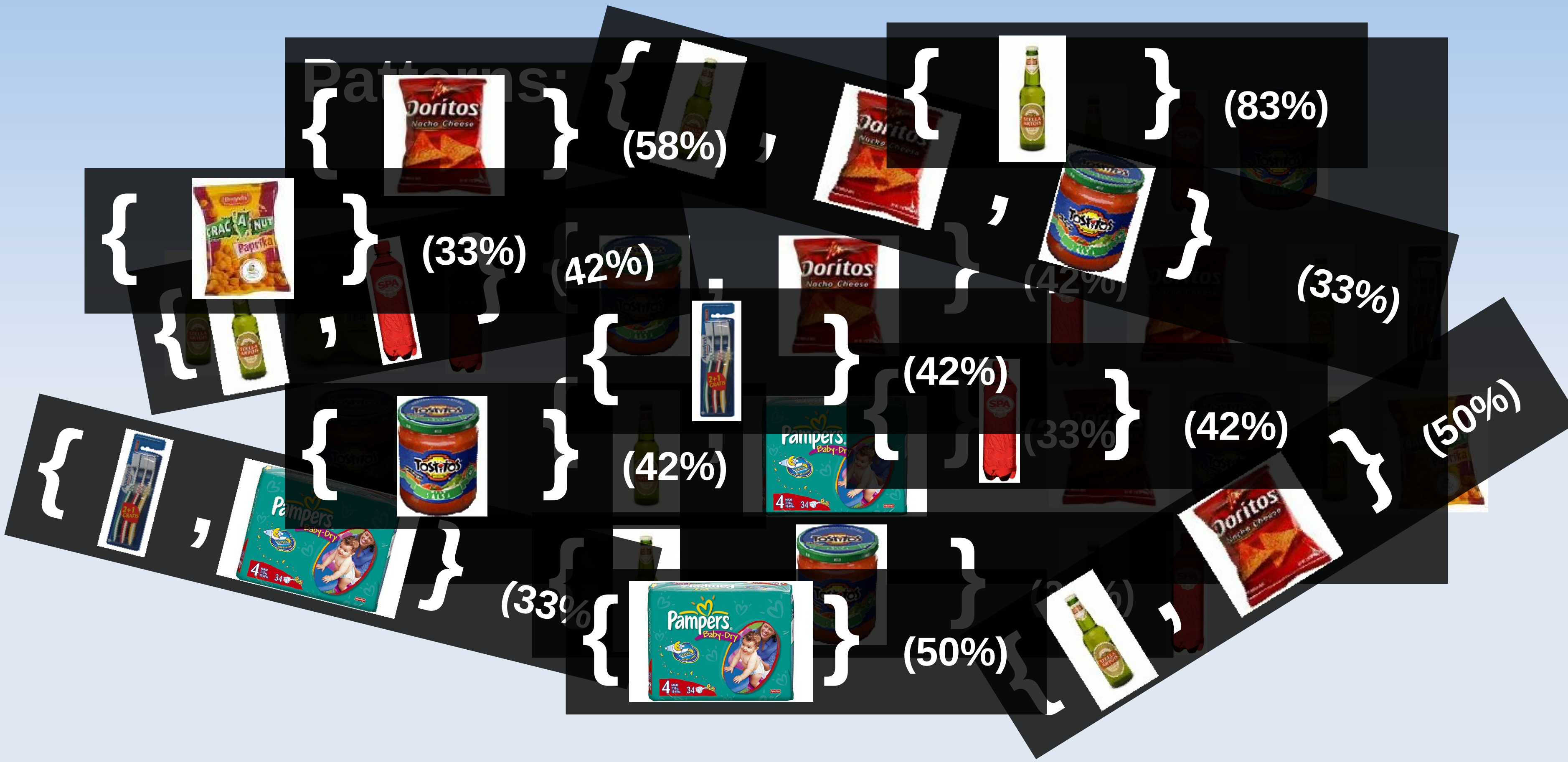




(frequent) Itemset mining

Transactions:

Too many patterns



- Time-consuming to interpret
- Long algorithmic runtime

Goal: find patterns in transactional data

Solution: Itemset Mining

Problem: too many patterns

Solution: Constraint-based Itemset Mining

⇒ select only interesting patterns, based on domain knowledge

Constraint-based mining

Use of constraints in data mining
to specify the desired set of solutions
(Mannila & Toivonen, 97)

$$Th(\mathcal{L}, Q, \mathcal{D}) = \{p \in \mathcal{L} | Q(p, \mathcal{D}) = true\}$$

- $\mathcal{L} = 2^{\mathcal{I}}$, i.e., itemsets — Pattern space Π
- $\mathcal{D} \subset \mathcal{L}$, i.e., transactions — Data X
- $Q(p, \mathcal{D}) = true$ if $freq(p, \mathcal{D}) \geq t$ — 0/1 Pattern strength

Constraint-based Itemset Mining

- condensed representations
  - Maximal patterns: remove all redundancy
  - Closed patterns: remove redundancy, keep frequencies
  - *delta*-closed patterns: closed + fault tolerance
- user defined constraints
  - human readable → size(*itemset*) ≤ 5
  - high value → total_cost(*itemset*) ≥ 100 £
  - infrequent on other dataset → freq_part2(*itemset*) ≤ 1%

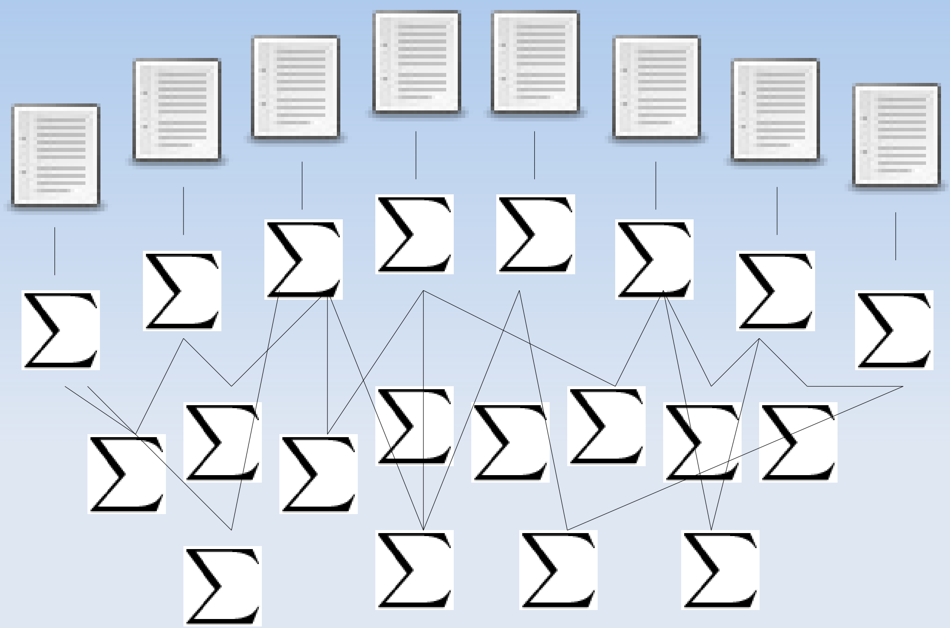
...

Constraint-based Itemset Mining (cont.)

+ many constraints proposed

- new constraint often require new implementations

- combining constraints ?



state-of-the-art =

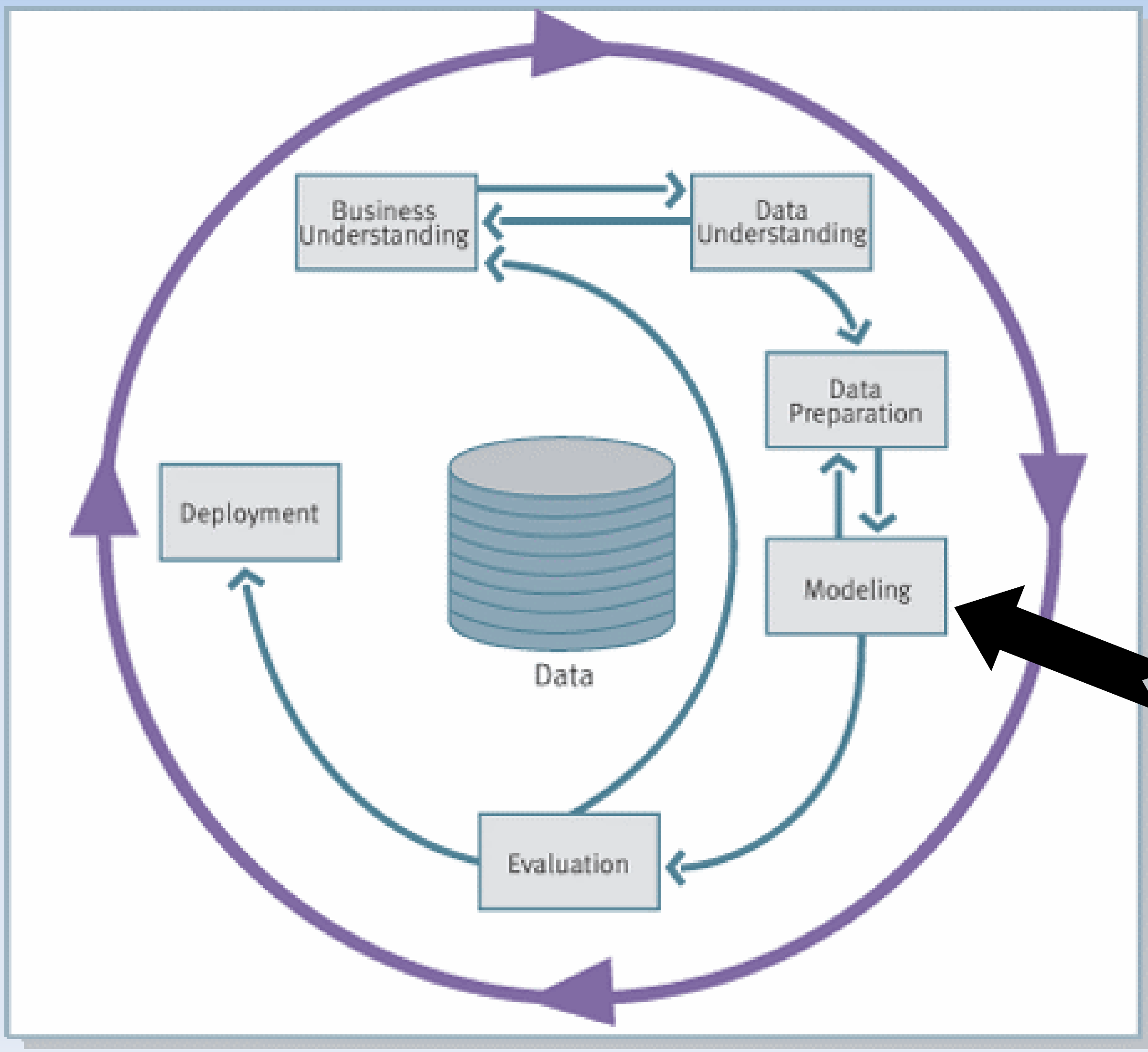
hard-coded support for some popular constraint families.

```
if (anti-monotone)
  then: ...
if (monotone)
  then: ...
if (convertible)
  then: ...
    if (convertible-anti-monotone)
      then: ...
    if (convertible-monotone)
      then: ...
if (weak-anti-monotone)
  then: ...
```

=> No principled approach

The need for a principled approach

The Data Mining process model:



Constraints

Constraint Programming for Itemset Mining

I. Motivation, constraint-based mining

II. Constraint Programming basics

III. Constraint-based itemset mining using CP

IV. Correlated itemset mining using CP

V. Conclusions.

Constraint programming:

- ... solves combinatorial satisfaction problems
- ... is used in many *applications*
- ... is an *active* research area
- ... is among the most *efficient* general problem solving techniques

How CP works

Constraint Programming =

MODEL (by user)

\+

SEARCH (by solver)

A CP model

- variables

  $[E_{11} \dots E_{99}]$

- domains

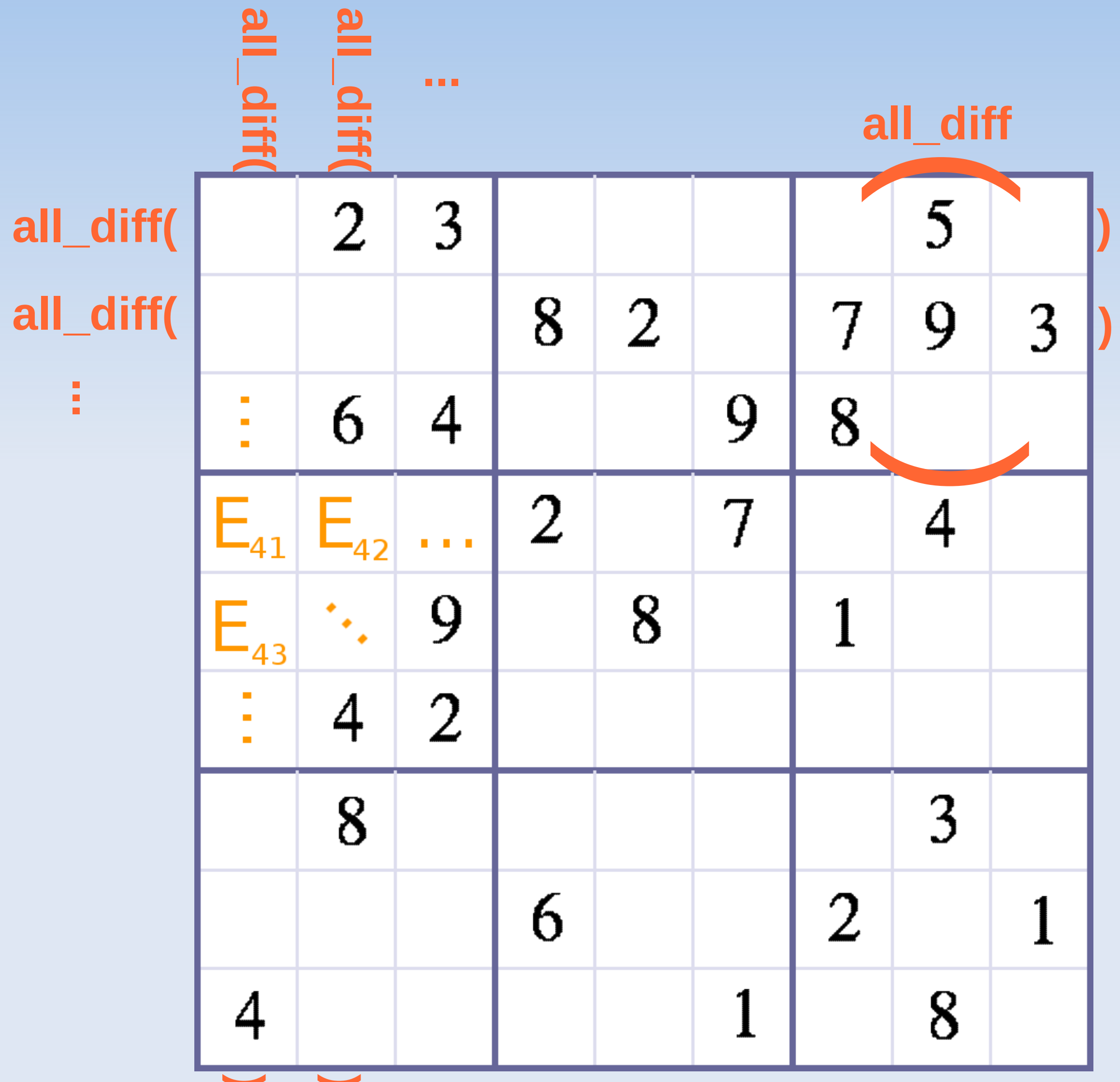
  $E_{xy} = \{1 \dots 9\}$

- constraints

  $\text{all\_different}([E_{1x}]), \dots$

  $\text{all\_different}([E_{x1}]), \dots$

  $\text{all\_different}([E_{11} \dots E_{33}]), \dots$



The CP Search

Two key principles:

- **Propagation** of constraints

  eg. alldiff(X,Y,Z) X={1},Y={1,2},Z={1,2,3,4} → Y={2},Z={3,4}

  Every constraint is implemented by a propagator.

- **Branch** over values of variables

  eg. Propagation at fixpoint → branch over Z={3}

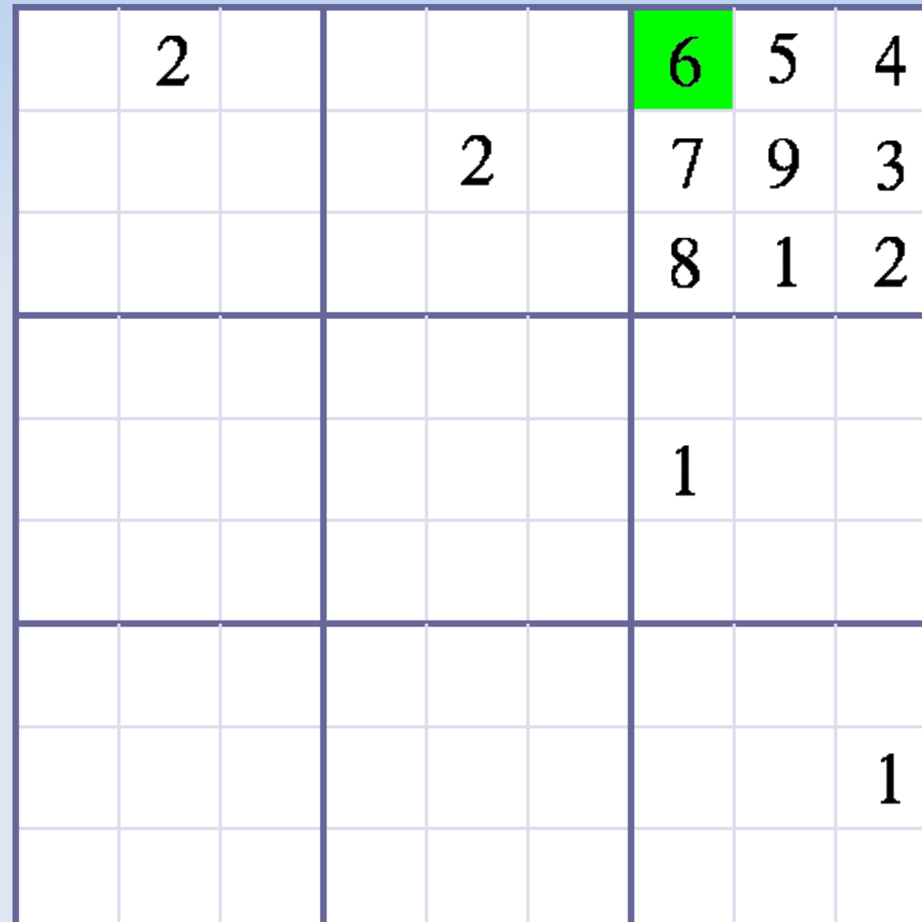
  Search is recursive and complete

A CP search

all rows: all_different(row)
all columns: all_different(col)
all squares: all_different(square)

CP: Branch & Propagate

- propagate 2 (row)
- branch 4
- propagate 6 (square)



Constraint Programming for Itemset Mining

I. Motivation, pattern mining

II. Constraint Programming basics

III. Constraint-based itemset mining using CP

IV. Correlated itemset mining using CP

V. More pattern mining at work

VI. Conclusion.

Constraint Programming

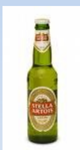
Surprisingly, Constraint Programming had not been used for constraint-based mining yet...

Constraint Programming for Itemset Mining
in short: (KDD2008)

- using **out-of-the-box** CP solvers
- allows to express **many** IM constraints
- easily **combine** all those constraints

Itemset mining

Transactions:






1)	0	1	1	1	0	0	0	0
2)	0	0	0	0	0	0	1	1
3)	1	0	0	0	1	1	0	0
4)	1	1	1	0	0	0	0	0
5)	1	0	0	1	0	1	0	1
6)	0	1	0	0	1	0	1	1
7)	1	0	0	1	0	0	1	1
8)	1	1	1	0	0	1	0	1
9)	1	1	0	0	0	1	1	0
10)	1	1	1	1	0	0	0	0
11)	1	0	0	0	0	1	1	1
12)	1	1	1	0	1	1	0	0

CP 4 IM

- variables

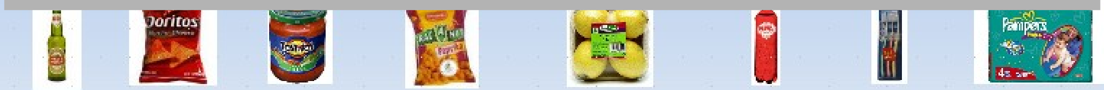
  $[I_1 \ldots I_n], [T_1 \ldots T_m]$

- domains

  $I_x, T_y = \{0, 1\}$

- constraints

  frequency: $\sum_{t \in T} T_t \geq \theta.$



$[T_1 \ldots T_{12}]$	$[I_1$	I_2	I_3	I_4	I_5	I_6	I_7	$I_8]$
1)	0	1	1	1	0	0	0	0
2)	0	0	0	0	0	0	1	1
3)	1	0	0	0	1	1	0	0
4)	1	1	1	0	0	0	0	0
5)	1	0	0	1	0	1	0	1
6)	0	1	0	0	1	0	1	1
7)	1	0	0	1	0	0	1	1
8)	1	1	1	0	0	1	0	1
9)	1	1	0	0	0	1	1	0
10)	1	1	1	1	0	0	0	0
11)	1	0	0	0	0	1	1	1
12)	1	1	1	0	1	1	0	0

CP 4 IM

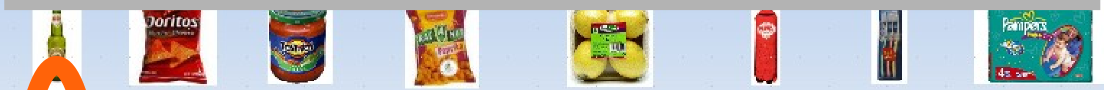
- variables

  $[I_1 \dots I_n], [T_1 \dots T_m]$

- domains

  $I_x, T_y = \{0, 1\}$

- constraints



[T1 … T12]	[I1	I2	I3	I4	I5	I6	I7	I8]
1)	0	1	1	1	0	0	0	0
2)	0	0	0	0	0	0	1	1
3)	1	0	0	0	1	1	0	0
4)	1	1	1	0	0	0	0	0
5)	1	0	0	1	0	1	0	1
6)	0	1	0	0	1	0	1	1
7)	1	0	0	1	0	0	1	1
8)	1	1	1	0	0	1	0	1
9)	1	1	0	0	0	1	1	0
10)	1	1	1	1	0	0	0	0
11)	1	0	0	0	0	1	1	1
12)	1	1	1	0	1	1	0	0

frequency: $\sum_{t \in \mathcal{T}} T_t \geq \theta.$

OR freq. reified: $\forall i \in \mathcal{I} : I_i = 1 \quad \rightarrow \sum_{t \in \mathcal{T}} T_t \mathcal{D}_{ti} \geq \theta.$

CP 4 IM

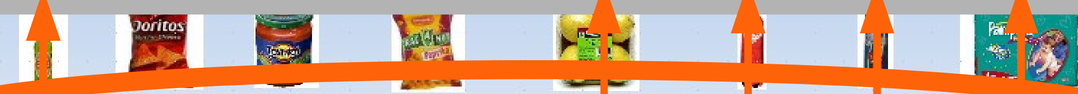
- variables

  $[I_1 \dots I_n], [T_1 \dots T_m]$

- domains

  $I_x, T_y = \{0, 1\}$

- constraints



[T_1 T_2 T_3 T_4 T_5 T_6 T_7 T_8 T_9 T_{10} T_{11} T_{12}]

	[I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8]
1)	0	1	1	1	0	0	0	0
2)	0	0	0	0	0	0	1	1
3)	1	0	0	0	1	1	0	0
4)	1	1	1	0	0	0	0	0
5)	1	0	0	1	0	1	0	1
6)	0	1	0	0	1	0	1	1
7)	1	0	0	1	0	0	1	1
8)	1	1	1	0	0	1	0	1
9)	1	1	0	0	0	1	1	0
10)	1	1	1	1	0	0	0	0
11)	1	0	0	0	0	1	1	1
12)	1	1	1	0	1	1	0	0

frequency: $\sum_{t \in \mathcal{T}} T_t \geq \theta.$

OR freq. reified: $\forall i \in \mathcal{I} : I_i = 1 \rightarrow \sum_{t \in \mathcal{T}} T_t \mathcal{D}_{ti} \geq \theta.$

\+ coverage: $\forall t \in \mathcal{T} : T_t = 1 \leftrightarrow \sum_{i \in \mathcal{I}} I_i (1 - \mathcal{D}_{ti}) = 0.$

Itemset Mining in CP (FIMCP)

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

```
given NrT, NrI : int
given TDB : matrix indexed by [int(1..NrT),int(1..NrI)] of int
given Freq : int

find Items : matrix indexed by [int(1..NrI)] of bool
find Trans : matrix indexed by [int(1..NrT)] of bool

such that

$ encode TDB: every ... ed Items
forall t: int(1..NrT).
  Trans[t] <=> ((sum i: int(1..NrI). Items[i]*(1-TDB[t,i])) = 0),

$ frequency: every Item ... Trans
forall i: int(1..NrI).
  Items[i] => ((sum t: int(1..NrT). Trans[t]*TDB[t,i]) >= Freq)
```

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

$$\forall i \in \mathcal{I} : I_i = 1 \quad \rightarrow \quad \sum_{t \in \mathcal{T}} T_t \mathcal{D}_{ti} \geq \theta.$$

The FIM_CP search

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

freq >= 2: $\forall i \in \mathcal{I} : I_i = 1 \quad \rightarrow \quad \sum_{t \in \mathcal{T}} T_t \mathcal{D}_{ti} \geq \theta.$

CP: Branch & Propagate

- propagate i2 (freq)

Intuition: infrequent

i2 can never be part of freq. superset

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

The FIM_CP search

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

freq >= 2: $\forall i \in \mathcal{I} : I_i = 1 \quad \rightarrow \quad \sum_{t \in \mathcal{T}} T_t \mathcal{D}_{ti} \geq \theta.$

CP: Branch & Propagate

- propagate i2 (freq)
- propagate t1 (coverage)

Intuition: unavoidable

t1 will always be covered

	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

The FIM_CP search

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

freq >= 2: $\forall i \in \mathcal{I} : I_i = 1 \quad \rightarrow \quad \sum_{t \in \mathcal{T}} T_t \mathcal{D}_{ti} \geq \theta.$

CP: Branch & Propagate

- propagate i2 (freq)
- propagate t1 (coverage)

		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

The FIM_CP search

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

freq >= 2: $\forall i \in \mathcal{I} : I_i = 1 \quad \rightarrow \quad \sum_{t \in \mathcal{T}} T_t \mathcal{D}_{ti} \geq \theta.$

CP: Branch & Propagate

- propagate i2 (freq)
- propagate t1 (coverage)
- branch i1=1
- propagate t3 (coverage)

Intuition: obsolete

t3 is missing an item of the itemset

		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

The FIM_CP search

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

freq >= 2: $\forall i \in \mathcal{I} : I_i = 1 \quad \rightarrow \quad \sum_{t \in \mathcal{T}} T_t \mathcal{D}_{ti} \geq \theta.$

CP: Branch & Propagate

- propagate i2 (freq)
- propagate t1 (coverage)
- branch i1=1
- propagate t3 (coverage)
- propagate i3 (freq)

		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

Intuition: infrequent

i3 can never be part of freq. superset

The FIM_CP search

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

freq >= 2: $\forall i \in \mathcal{I} : I_i = 1 \quad \rightarrow \quad \sum_{t\in\mathcal{T}} T_t \mathcal{D}_{ti} \geq \theta.$

CP: Branch & Propagate

- propagate i2 (freq)
- propagate t1 (coverage)
- branch i1=1
- propagate t3 (coverage)
- propagate i3 (freq)
- propagate t2 (coverage)

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

The FIM_CP search

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

freq >= 2: $\forall i \in \mathcal{I} : I_i = 1 \quad \rightarrow \quad \sum_{t \in \mathcal{T}} T_t \mathcal{D}_{ti} \geq \theta.$

CP: Branch & Propagate

- propagate i2 (freq)
- propagate t1 (coverage)
- branch i1=1
- propagate t3 (coverage)
- propagate i3 (freq)
- propagate t2 (coverage)
- ...

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

FIM_CP model: expressive

- Base model (Frequent Itemset Mining)

$$T_t = 1 \Leftrightarrow \sum_i I_i (1 - D_{ti}) = 0$$
$$I_i = 1 \Rightarrow \sum_t T_t D_{ti} \geq Freq$$

- Maximal Frequent Itemset Mining

$$T_t = 1 \Leftrightarrow \sum_i I_i (1 - D_{ti}) = 0$$
$$I_i = 1 \Leftrightarrow \sum_t T_t D_{ti} \geq Freq$$

- Closed Itemset Mining

$$T_t = 1 \Leftrightarrow \sum_i I_i (1 - D_{ti}) = 0$$
$$I_i = 1 \Rightarrow \sum_t T_t D_{ti} \geq Freq$$
$$I_i = 1 \Leftrightarrow \sum_t T_t (1 - D_{ti}) = 0$$

- δ-Closed Itemset Mining

$$T_t = 1 \Leftrightarrow \sum_i I_i (1 - D_{ti}) = 0$$
$$I_i = 1 \Rightarrow \sum_t T_t D_{ti} \geq Freq$$
$$I_i = 1 \Leftrightarrow \sum_t T_t (1 - \delta - D_{ti}) = 0$$

FIM_CP model: general

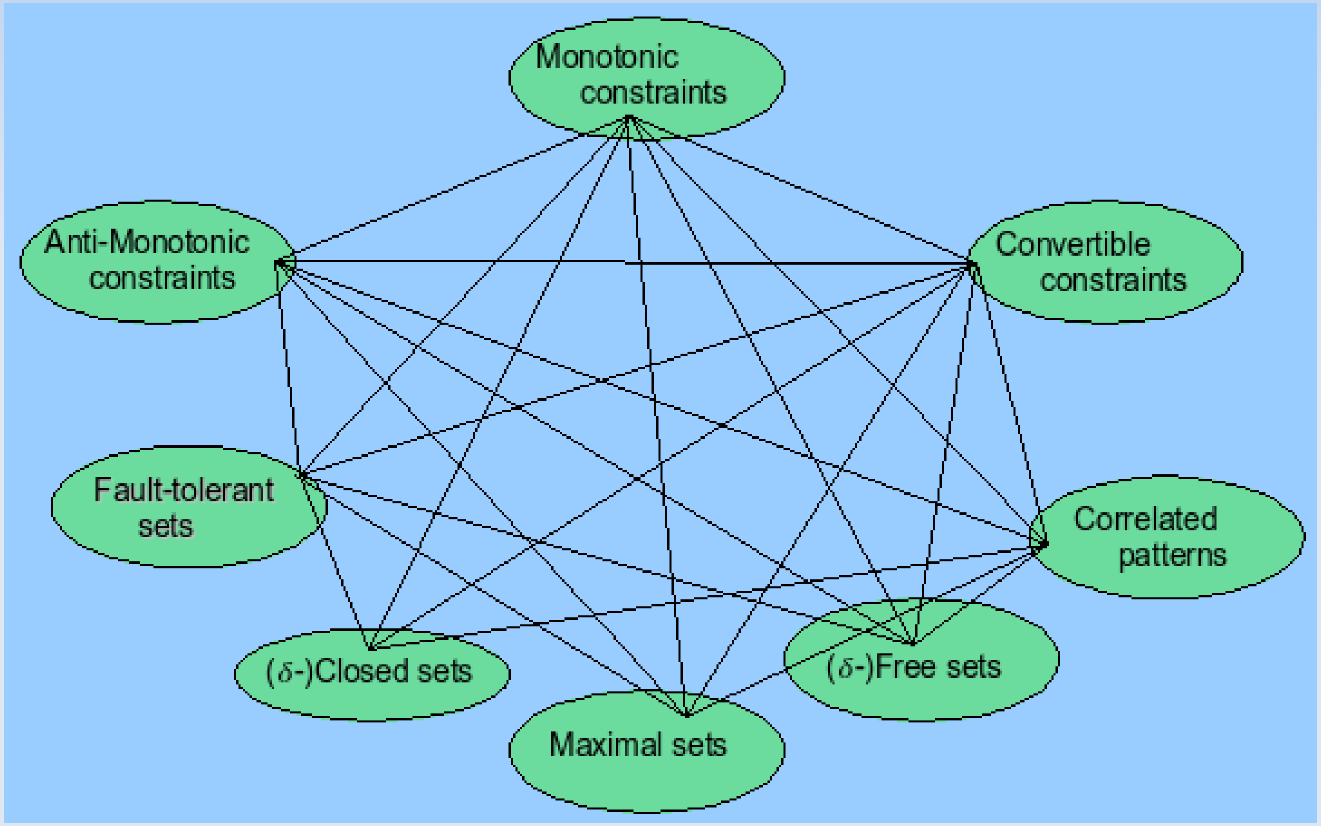
	LCM [15]	MAFIA [6]	ExAMiner [4]	DualMiner [5]	CP
Constraints on data					
Minimum frequency	X	X	X	X	X
Maximum frequency				X	X
Emerging patterns					X
Condensed Representations					
Maximal	X	X		X	X
Closed	X	X			X
$\delta-$Closed					X
Constraints on syntax					
Max/Min total cost			X	X	X
Minimum average cost			X		X
Max/Min size	X	X	X	X	X

Table 1: Comparison of Itemset Miners

=> most general system to date !

FIM_CP model: flexible

combining constraints is the core of CP

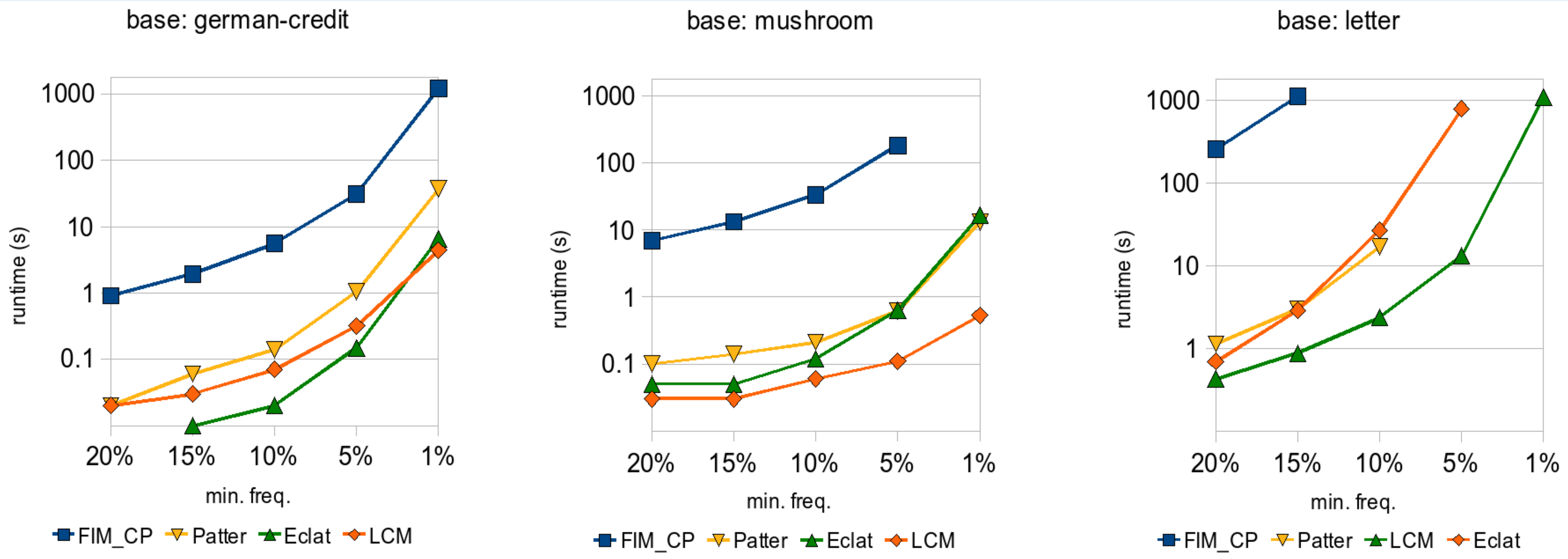


=> most flexible system to date !

In Short: FIM_CP

- Principled approach
- Using generic Constraint Programming
- Declarative language, very expressive

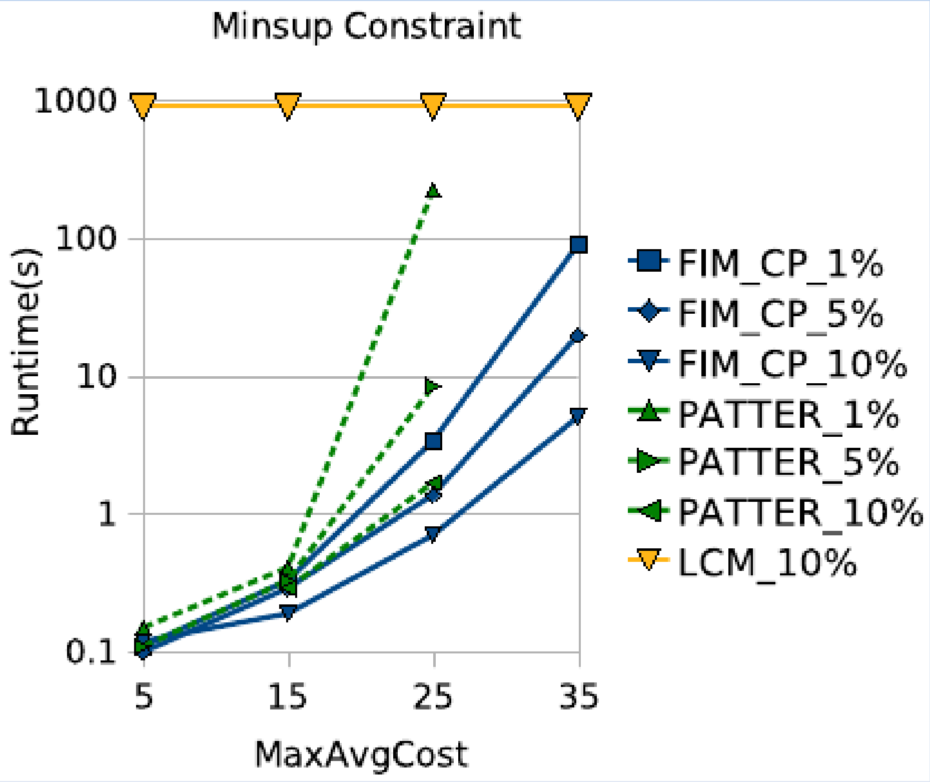
Runtime behavior, unconstrained



Dataset properties:

	german-credit	mushroom	letter
# items	77	116	74
# transactions	1000	8124	20000
sparseness	0.28	0.17	0.33

Runtime behavior, constrained



Dataset: segment 61x2310 (sparseness: 0.51)

patterns with min. freq. of 10% only: > 64 million
Impossible to mine unconstrained with lower freq. treshold.

Experiment conclusions

bad for

- very large datasets (> 1.000.000 transactions)
- very low frequency unconstraint (< 0.1 %)

ideal for

- studying existing constraints
- rapid prototyping of new constraints
- exploratory constraint-based mining

Constraint Programming for Itemset Mining

I. Motivation, pattern mining

II. Constraint Programming basics

III. Constraint-based itemset mining using CP

IV. Correlated itemset mining using CP

V. Conclusions.

Correlated Itemset Mining

Contingency Table

TP: 3 (=p)	FP: 0 (=n)	3
FN: 1	TN: 3	4
P: 4	N: 3	

Constraint-based mining

- Frequent itemset mining (association rule mining)

  - Traditional pattern mining:
    $Th(\mathcal{L}, Q, \mathcal{D}) = \{p \in \mathcal{L} | Q(p, \mathcal{D}) = true\}$

- Correlated itemset mining (correlation rule mining)

  - Correlated pattern mining with function $\phi(p, \mathcal{D})$, (χ^2),
    $Th(\mathcal{L}, Q, \mathcal{D}) = \arg_{p \in \mathcal{L}} \max_k \phi(p, \mathcal{D})$

Correlated itemset mining

Also known as:

- **Discriminative itemset mining**
- Contrast set mining
- Emerging itemsets
- Subgroup discovery
- Interesting itemsets

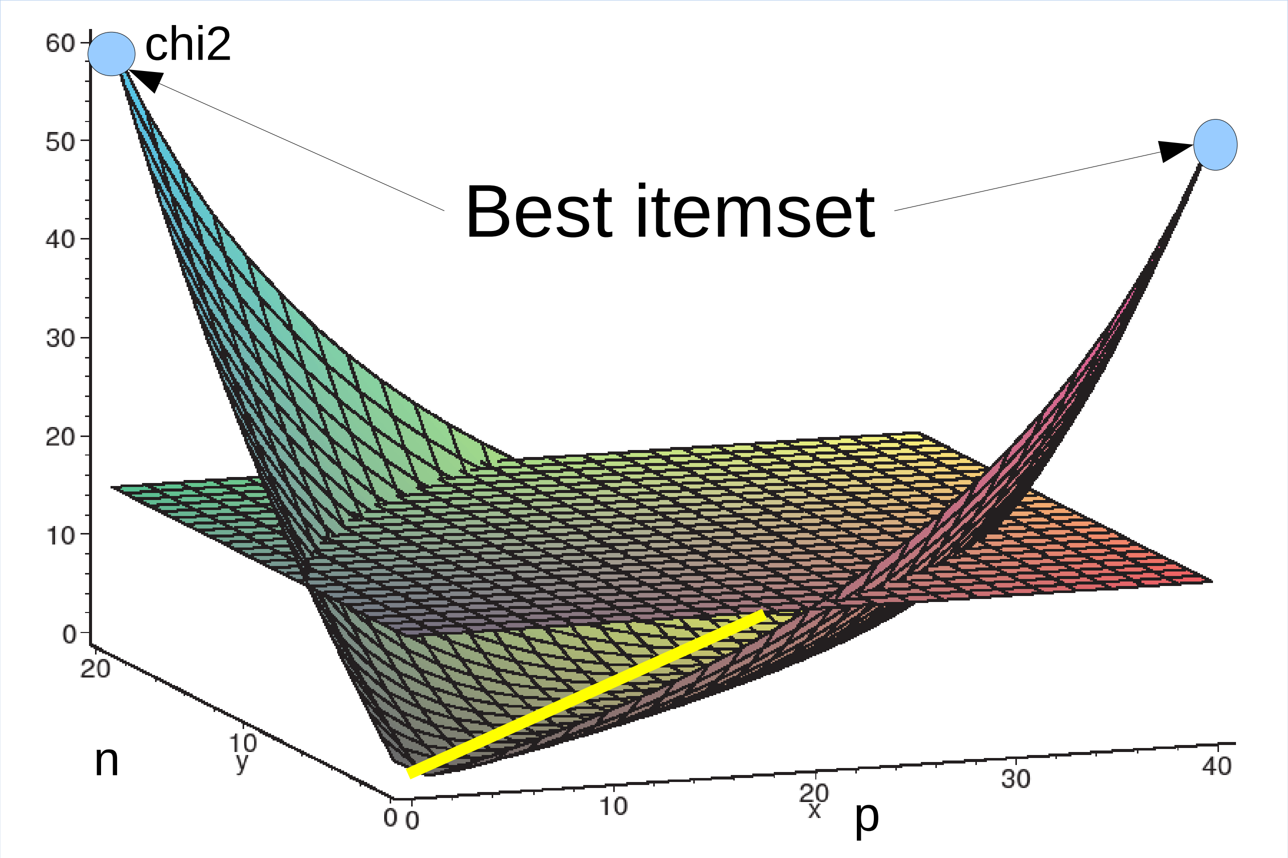
They all find an itemset/rule in labeled data that optimises a convex (correlation) measure.

ROC analysis: PN-space

Contingency Table

TP: 3 (=p)	FP: 0 (=n)	3
FN: 1	TN: 3	4
P: 4	N: 3	

Measuring correlation



Many correlation functions (chi2, fisher, inf. gain) are convex and zero on the diagonal

Convex measures in CP

- Frequent itemset mining:

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

  frequency: $\forall i \in \mathcal{I} : I_i = 1 \quad \rightarrow \quad \sum_{t \in \mathcal{T}} T_t \mathcal{D}_{ti} \geq \theta.$

- Correlated itemset mining:

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

  correlation: $\forall i \in \mathcal{I} : I_i = 1 \quad \rightarrow \quad f(\sum_{t \in \mathcal{T}^+} T_t \mathcal{D}_{ti}, \sum_{t \in \mathcal{T}^-} T_t \mathcal{D}_{ti}) \geq \theta$

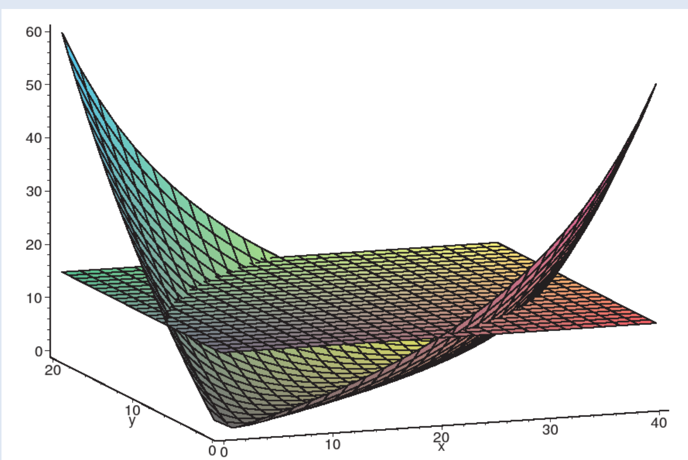
  + branch and bound search

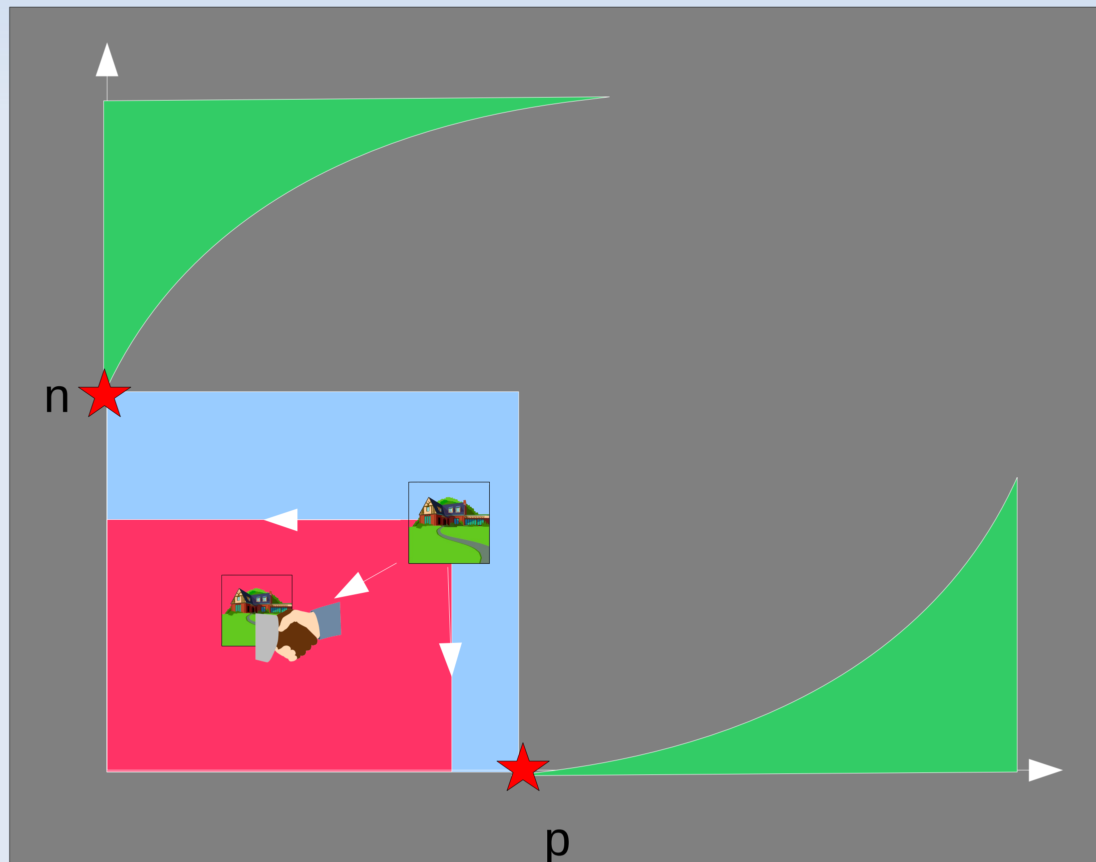
Bound in PN-space

Morishita & Sese, 2000

General to specific search

- Adding an item will give equal or lower *p* and *n*

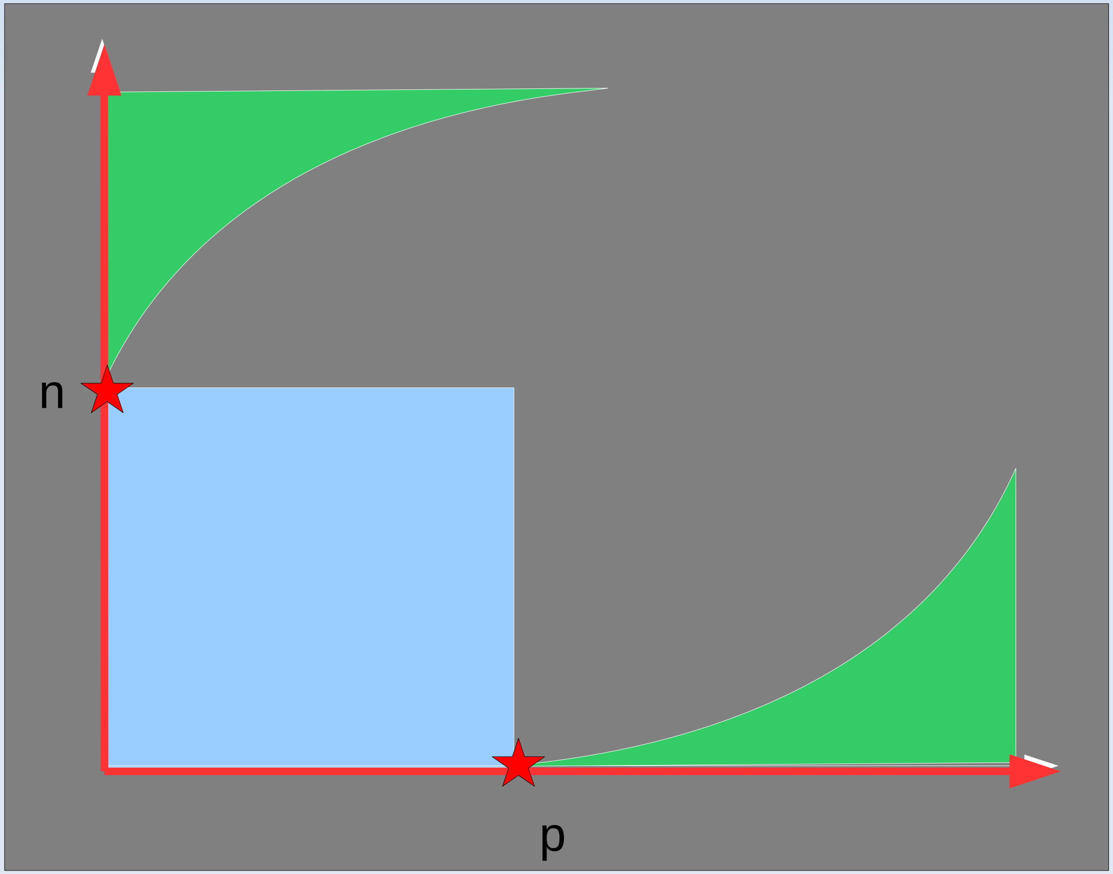




Improved bound in PN-space

Key observation: unavoidable transactions

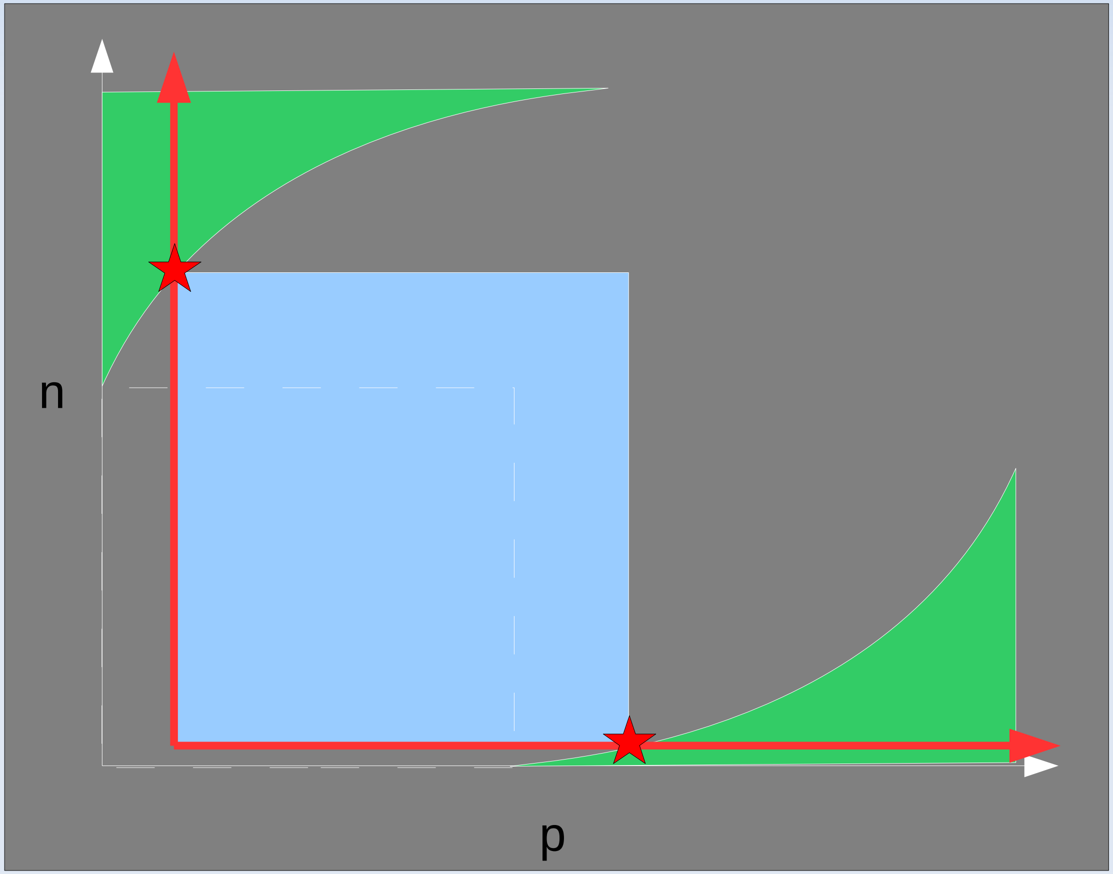
	i1 0/1	i2 0	i3 0/1	i4 0/1
t1 0/1	1	0	1	1
t2 0/1	1	1	0	1
t3 0/1	0	0	1	1



Better bound in PN-space

Key observation: unavoidable transactions

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

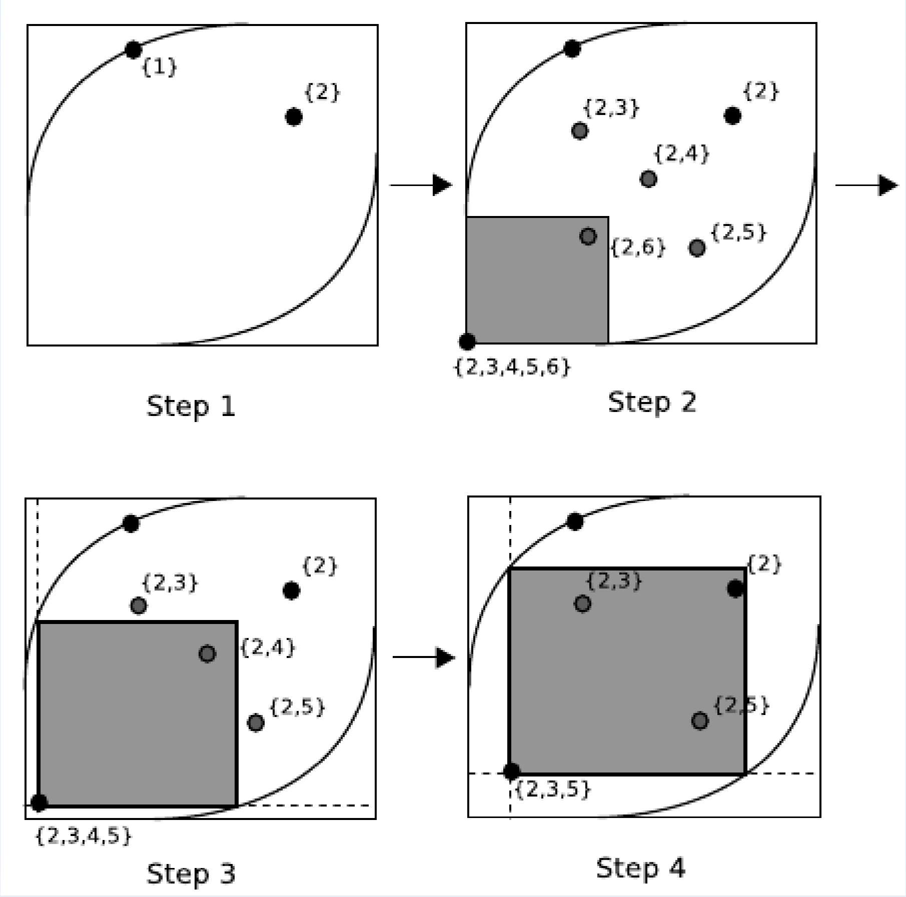


Branch and propagate CIMCP

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

correlation: $$\forall i \in \mathcal{I} : I_i = 1 \quad \rightarrow \quad f(\sum_{t \in \mathcal{T}^+} T_t \mathcal{D}_{ti}, \sum_{t \in \mathcal{T}^-} T_t \mathcal{D}_{ti}) \geq \theta$$

iterative pruning loop:



Correlation measures

Taking the *unavoidable* transactions into account, results in more effective pruning...

Correlated Itemset Mining in ROC space:
A Constraint Programming Approach

in short: (KDD2009)

- based on principles of **ROC analysis**
- using insights from **Constraint Programming**
- very **fast and effective pruning**

Experiments

- Branch and bound search for top-*1* pattern
- In CP:
  - 1-support (traditional minimum support)
  - 2-support (Morishita & Sese, 2000)
  - 4-support (with unavoidable transactions)

Experiments in CP

Runtime in seconds, >900s indicated by >

Name	Density	4-supp.	2-supp.	1-supp.
anneal	0.45	0.22	24.09	72.71
australian-credit	0.41	0.30	0.63	17.52
breast-wisconsin	0.5	0.28	13.66	228.08
diabetes	0.5	2.45	128.04	>
german-credit	0.34	2.39	66.79	>
heart-cleveland	0.47	0.19	2.15	29.58
hypothyroid	0.49	0.71	10.91	>
ionosphere	0.5	1.44	>	>
kr-vs-kp	0.49	0.92	46.20	713.35
letter	0.5	52.66	>	>
mushroom	0.18	14.11	13.48	27.31
pendigits	0.5	3.68	>	>
primary-tumor	0.48	0.03	0.13	0.85
segment	0.5	1.45	>	>
soybean	0.32	0.05	0.07	0.38
splice-1	0.21	30.41	31.11	35.02
vehicle	0.5	0.85	>	>
yeast	0.49	5.67	781.63	>

Experiments

- Outside CP:
  - DDPMine [ICDE'08]
  - LCM (FIMI's “winner”)
  - CIMCP (4-bound in Gecode CP solver)
  - corrmine (4-bound pruning implemented in a eclat-like specialised miner)

Experiments in CP

Runtime in seconds, >900s indicated by >
memory exhausted by -

Name	corrmine	cimcp	ddpmine	lcm
anneal	0.02	0.22	22.46	7.92
australian-credit	0.01	0.30	3.40	1.22
breast-wisconsin	0.03	0.28	96.75	27.49
diabetes	0.36	2.45	–	697.12
german-credit	0.07	2.39	–	30.84
heart-cleveland	0.03	0.19	9.49	2.87
hypothyroid	0.02	0.71	–	>
ionosphere	0.24	1.44	–	>
kr-vs-kp	0.02	0.92	125.60	25.62
letter	0.65	52.66	–	>
mushroom	0.03	14.11	0.09	0.03
pendigits	0.18	3.68	–	>
primary-tumor	0.01	0.03	0.26	0.08
segment	0.06	1.45	–	>
soybean	0.01	0.05	0.05	0.02
splice-1	0.05	30.41	1.86	0.02
vehicle	0.07	0.85	–	>
yeast	0.80	5.67	–	185.28
avg. when found:	*0.15*	*6.55*	*28.88+*	*81.54+*

Experiment conclusion

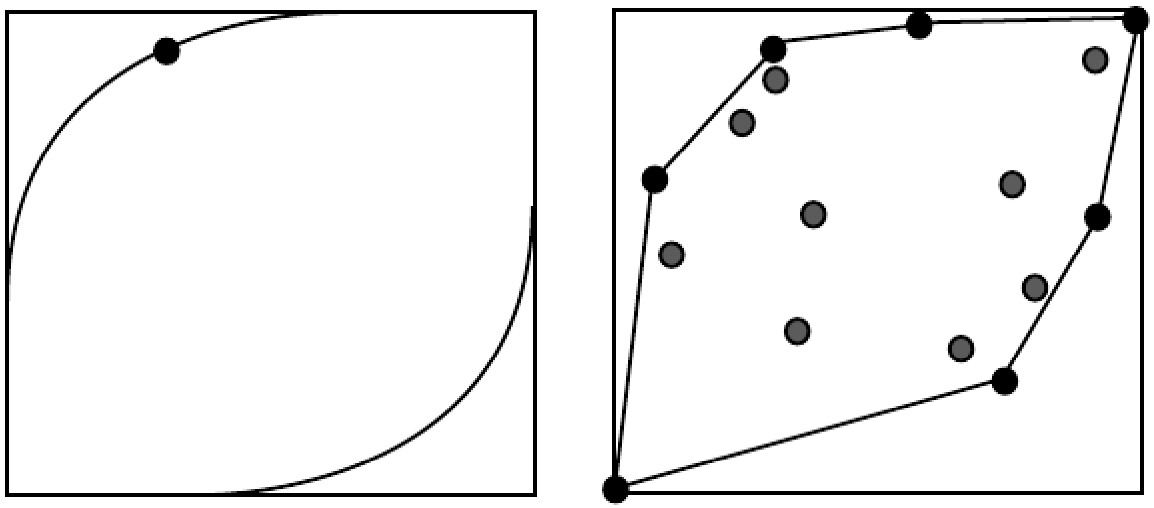
- New bound results in far **better pruning**
- CP (gecode) incurs **overhead** for very sparse datasets
- Principles from CP-mining **carry back over** to traditional mining algorithms
- **Fastest algorithm** in all our experiments

Parameter-free mining ?

Can we do even better ?

- Mine all possible itemsets for which a correlation measure exists under which it is optimal ?

= All itemset on the convex hull in ROC space



Isometric of convex correlation score

Convex hull of itemsets

Experiments convex hull

Name	cimcp time (s)	cimcp convex hull time (s)	cimcp convex hull size of hull
anneal	0.22	0.44	17
australian-credit	0.30	1.33	22
breast-wisconsin	0.28	0.83	20
diabetes	2.45	11.9	30
german-credit	2.39	3.93	21
heart-cleveland	0.19	0.37	20
hypothyroid	0.71	3.01	19
ionosphere	1.44	8.69	15
kr-vs-kp	0.92	1.75	17
letter	52.66	405.14	34
mushroom	14.11	32.45	10
pendigits	3.68	45.79	19
primary-tumor	0.03	0.07	16
segment	1.45	8.96	6
soybean	0.05	0.09	9
splice-1	30.41	40.13	10
vehicle	0.85	4.12	22
yeast	5.67	25.51	28
average:	*6.55*	*33.03*	*18.61*

- No parameters
- All patterns on convex hull
- Possible !

- Reasonably small hulls
- Reasonable increase in runtime for entire hull

Constraint Programming for Itemset Mining

I. Motivation, pattern mining

II. Constraint Programming basics

III. Constraint-based itemset mining using CP

IV. Correlated itemset mining using CP

V. **Conclusions.**

Unrelated work

Boosting / sparsity induced learning

- Every correlated itemset is a rule; a weak classifier
- LPboost [iboost: H. Saigo, T. Uno, K. Tsuda, 2007]

Statistical validation of itemsets

- *Geoffrey I. Webb:* Discovering significant patterns. *Machine Learning Journal (2008)*
- *Arianna Gallo, Tijl De Bie, Nello Cristianini:* MINI: Mining Informative Non-redundant Itemsets. *PKDD 2007*
- *Sami Hanhijärvi, Markus Ojala, Niko Vuokko, Kai Puolamäki, Nikolaj Tatti, Heikki Mannila:* Tell me something I don't know: randomization strategies for iterative data mining. *KDD 2009*

Constraint Programming for Itemset Mining

A **new methodology** for constraint-based mining

- Pattern Mining as model + search
- Using a declarative CP language
- Itemset Mining as standard depth-first search

Yet keeping the **existing principles**.

- Anti-monotonicity
- Similar traversal as specialized miners like
  eclat, dual miner, mafia, examiner, ...

Constraint Programming for Itemset Mining

Many additional advantages:

- Easily **combining** constraints
  - Demonstrated: Emerging + delta-closed + max-size + min-size
- **Studying** constraints independently
  - Demonstrated: Correlation constraint; 1-bound, 2-bound and 4-bound
- Rapid **prototyping** of new constraints
  - Demonstrated: Entire ROC convex hull

Constraint Programming for Itemset Mining

CP4IM

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

→ http://www.cs.kuleuven.be/~DTAI/CP4IM ←

Challenges

CP (gecode) has overhead for sparse data

- Specialised solver with same flexibility ?

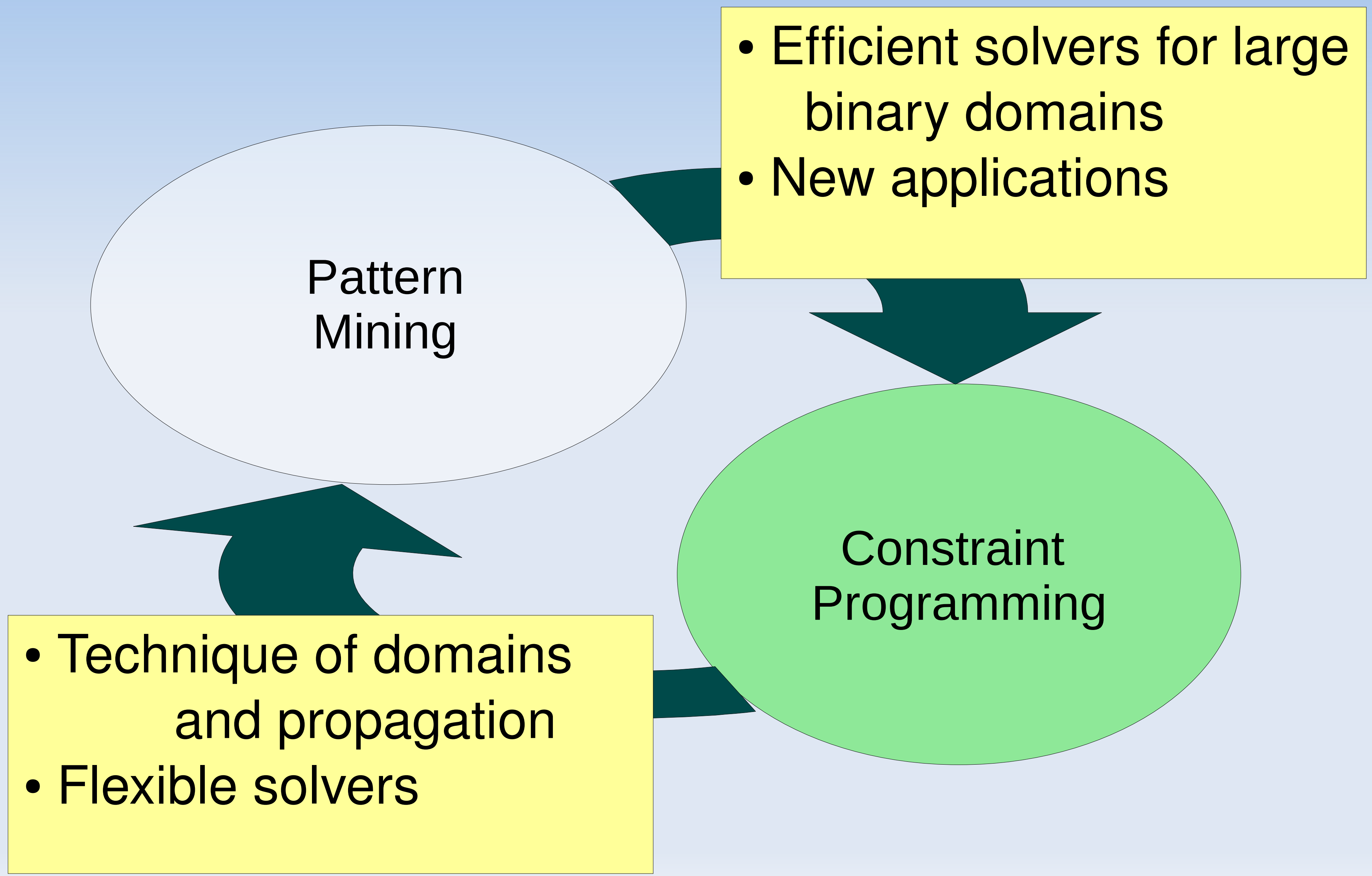
Building global models *(eg. boosting)*

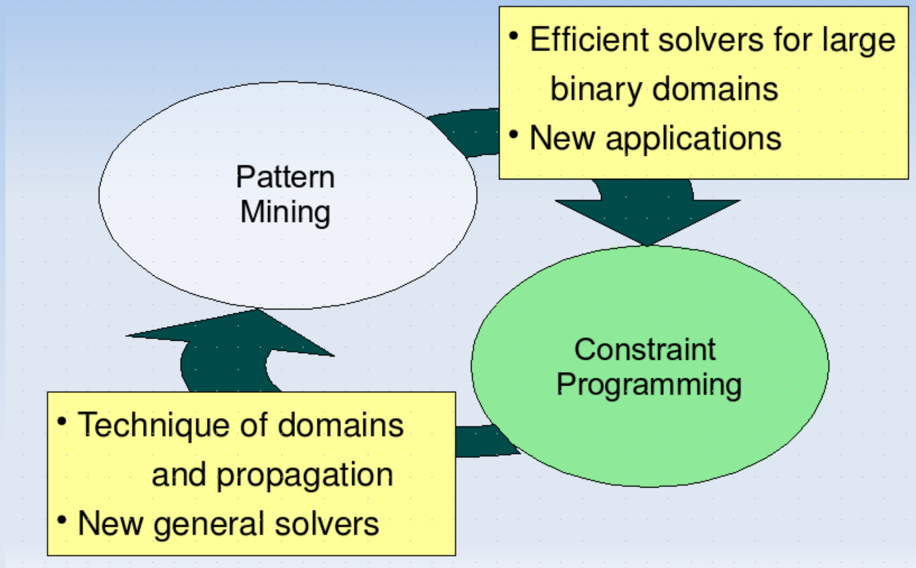
- Incorporate more of the learning in the mining ?

In Data Mining, different pattern types and data

- graphs, trees, sequences with CP ?

Bigger picture





questions ?

http://
www.cs.kuleuven.be/
~dtai/CP4IM/