Data Mining and Machine Learning with CP/SAT/MIP

Slides at:

https://sites.uclouvain.be/cp4dm/tutorial/ecmlpkdd19/

Siegfried Nijssen
Université catholique de Louvain, Belgium
siegfried.nijssen@uclouvain.be

Tias Guns
VUB Brussel and KU Leuven, Belgium
tias.guns@vub.be

Ian Davidson
University of California – Davis
davidson@cs.ucdavis.edu

Acknowledgments

- We gratefully accepted donations of slides from
  - Bich Duong, Christel Vrain, University d'Orleans
  - Francesco Bonchi, Yahoo!
  - John Aoga, UC Louvain
- Grants
  - Ian
    - National Science Foundation − "Explaining Unsupervised Learning − Combinatorial Optimization Approaches … "
    - Google − "Combining Symbolic Reasoning and Deep Learning: A Constrained Optimization Formulation"

Logistics

- Lots of hyper-links in the Slide PDF to papers, meetings etc.
- We will have a half an hour coffee break at 10:30pm.
  - Three parts each of approximately an hour
  - Solvers, Pattern Mining (Tias)
  - Pattern Mining, Decision trees (Siegfried)
  - Clustering (Ian)
- List of references in the slides after each part.
- We are here for most of the conference
  - Feel free to ask general clarifying question.

Why We Are Here

A growing interest in using Constraint Solvers (CP/SAT/MIP) in ML and DM

- Meetings
  - Dagstuhl 11201 (2011): Constraint Programming meets Machine Learning and Data Mining
  - Dagstuhl 14411 (2014): Constraints, Optimization and Data
- Workshops
  - CoCoMile 2012 ECAI, CoCoMile 2013 AAAI
- Journal Special Issue
  - AIJ - Combining Constraint Solving, Mining & Learning 2017
- CP 2017, 2018, 2019
  - Special track on ML/DM + CP
- IJCAI 2017
  - Tutorial
- Two dozen+ papers in the last five years on the topic at:
  - NIPS, IJCAI, AAAI, KDD, ICDM, ICML, SDM etc.
  - Best paper award at CP this year on ML
  - Too many to cover in this tutorial − apologies if we did not cover your work.

Purpose of This Tutorial

- Give the audience an overview of the variety of work being completed at the **one-way** intersection of CP/SAT/MIP **used** for ML and DM
- Three Broad Categories of Work
  - Pattern Mining
  - Clustering
  - Emerging domains
- But why do we need constraint based formulations?

Lots of Innovative ML/DM Formulations and Settings

Setting

- 1970s/80s
  - Single label, single set of features, learn a single model
- 1990s
  - Multi-view, multi-class, multi-label, multiple models
- 2000s
  - Transfer, semi-supervised active, structured learning, side-information
- 2010s
  - Big and Complex Data

Solver

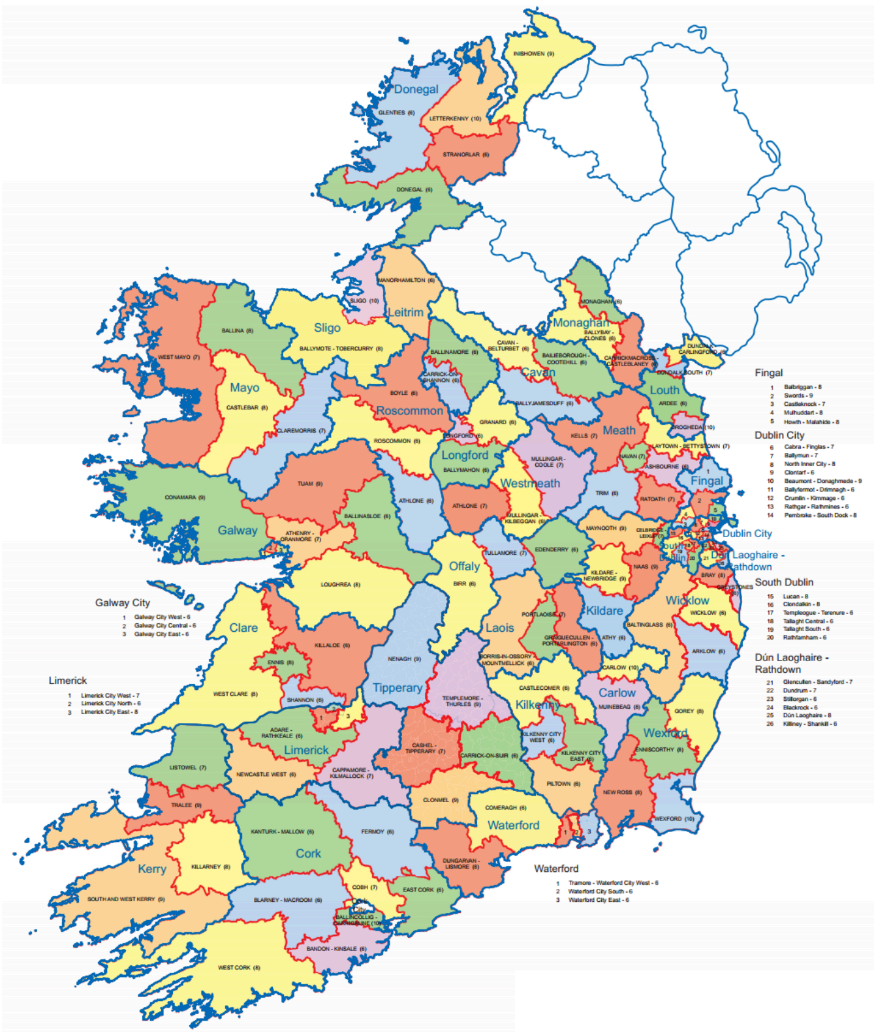
- 1970s/80s
  - Algorithms written in procedural languages
    - i.e. Decision trees, rule inducers
- 1990s
  - More complex algorithms but still procedural
    - i.e. Ensemble methods
- 2000s
  - Mathematical programming formulations
    - SVMs, Spectral Clustering
- 2010s
  - Deep Learning

Why SAT/CP/ILP for Optimization Formulated ML and DM?

- Despite the great work in ML and DM several core compromises and limitations exist
  - Modeling compromise
    1. Simple (typically linear) constraints
    2. Single mathematically convenient objective
  - Solving compromise
    1. Finds single local minima
    2. Relaxation to a continuous optimization problem (many problems are naturally discrete optimization)
- SAT/CP/MIP can alleviate these limitations and more
  - Easy to model and explore variations of problems
  - Constraints as a basis for a dialog b/w humans and machines.

Example - 1

[Kotthoff, Lars, et al. "Complex Clustering Using Constraint Programming: Modelling Electoral Map." (2015).]



Modeling compromise

1. Simple (typically linear) constraints
2. Single mathematically convenient objective

Solving compromise

1. Finds single local minima
2. Relaxation to a continuous optimization problem

How Are SAT/MIP/CP Solvers Being Used in ML/DM Settings?

Setting	Classic Setting	Extending Classic Settings	Completely New Settings
Formulation	Of the Shelf (Pure) SAT/ MIP/CP Formulation	Extensions To SAT/MIP/CP	Hybrid Formulations

Background on

SAT/MIP/CP

How Are SAT/MIP/CP Solvers Being Used in ML/DM Settings?

Pure SAT/MIP/CP Formulation

Extensions To SAT/MIP/CP Solvers

Hybrid Formulations

Pure SAT/MIP/CP Formulation:
declarative programming

Declarative programming

Imperative:

- Describe solution method: the <u>how</u>
- *ex: click once on the right-pointing arrow*

Declarative:

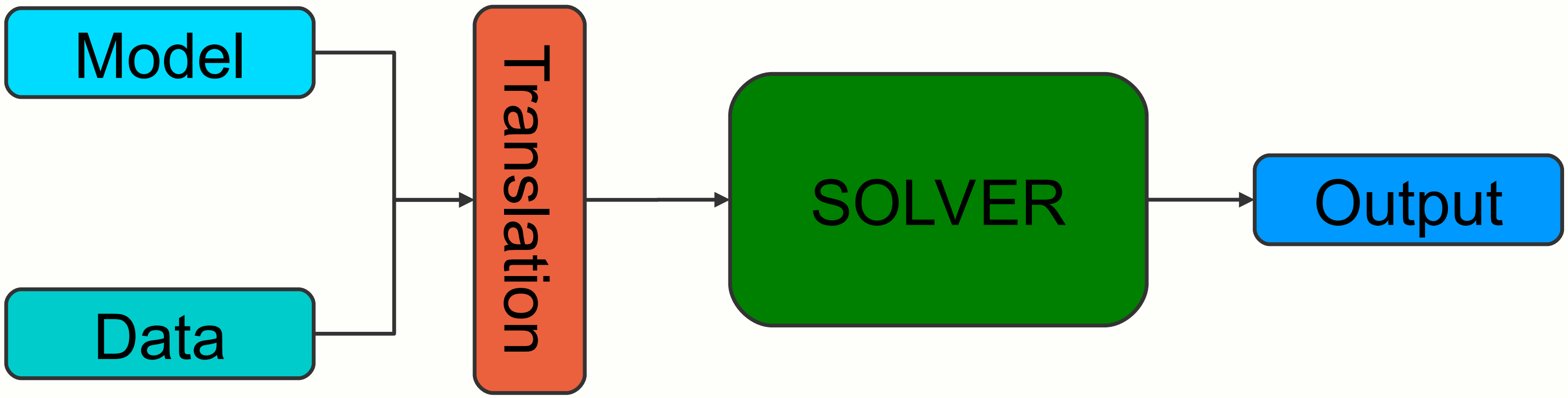
- Describe problem: the <u>what</u>
- *ex: go to the next slide*

Constraint solving: methods

"Combinatorial problem = <u>Model</u> + <u>Solve</u>"

Model = specification of constraints over variables

Solve = search for satisfying/optimal solutions



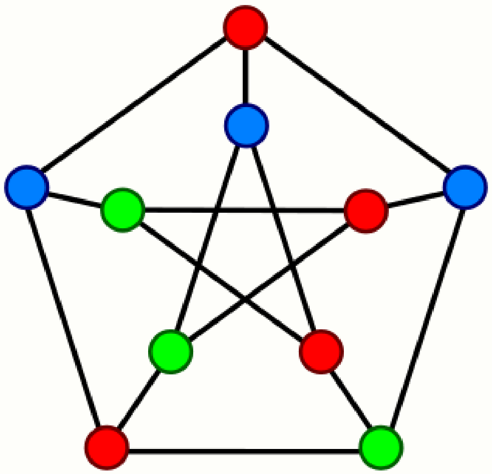
Many generic and efficient solvers available!

Modeling

	SAT Boolean Satisfiability	**MIP** Mixed Integer Programming	**CP** Constraint Programming
Variables	Boolean (True/False)	Boolean (0/1) Finite Integer Continuous	Boolean Finite Integer Continuous Set String, ...
Constraints	Logical (And/Or)	Linear (ext: quadratic)	Logical, Linear, Quadratic, Arbitrary…
Objective	/ (ext: MaxSAT)	Linear (ext: quadratic)	Linear, Quadratic, Arbitrary…

More expressive

Modeling example: graph coloring



- Given: a graph G=(V,E) with vertices V and edges E
- Find: a coloring of vertices V with minimal nr. of colors
  such that all (v_1,v_2) in E: $color(v_1) \neq color(v_2)$

Graph coloring: CP

- Variables: $X_1 .. X_n$ *one for each vertex*
  nr_colors *the number of colors*
- Domains: $D(X_i) = 1..n$ *colors numbered from 1 to n*
  D(nr_colors) = 1..n
- Minimize: nr_colors
- Constraints:
  - forall i,j in Edges: $X_i \,!= X_j$ *neighbors*
  - forall i: X_i < nr_colors *color count*
  - *optional: symmetry breaking*

Graph Coloring: MIP

- Variables: X_{ih} *X_{ij}: vertex i has color h*

  Y_h *Y_h: color h is used*

- Minimize: $\text{sum}_h\ Y_h$
- such that:
  - X_{ih} in {0,1}
  - Y_h in {0,1}
  - forall i: $\text{sum}_h\ X_{ih} = 1$ *each vertex one color*
  - forall i,j in Edges: $X_{ih} + X_{ih} <= Y_h$ *color use and neighbors*
  - *optional: symmetry breaking*

Graph coloring: SAT

<u>Satisfaction:</u> does a *k*-coloring exist?

<u>Variables:</u> X_{ih} *X_{ij}: vertex i has color h*

<u>such that:</u>

- forall i: OR(h in 1..n) X_{ih} *each vertex a color*
- forall i, forall h,g: NOT(X_{ih} AND X_{ig})
  each vertex no two colors
- forall i,j in Edges: NOT(X_{ih} AND X_{ih}) *neighbors*
- *optional: symmetry breaking*

Modeling differences

- CP: high level, problem structure more explicit

- MIP: low level, *relaxable* and as linear constraints

  (some modeling support in commercial solvers)

- SAT: low level, often need to write your own clause generators

Modeling: practical considerations

- Model size: MIP/SAT formulations can grow very large (millions of constraints)
- Modeling alternatives:
  - often different ways of modeling same (sub)problem
  - modeling choices matter, needs to be chosen experimentally
- Symmetric solutions and symmetry breaking

How Are SAT/MIP/CP Solvers Being Used in ML/DM Settings?

Pure SAT/MIP/CP Formulation

Extensions To SAT/MIP/CP Solvers

Hybrid Formulations

Improving solving by <u>exploiting problem structure</u>:

- CP: global constraints
- MIP: cutting planes
- *SAT: SAT Module theories (not covered here)*

→ use specialized algorithms to solve a specific subproblem

Extending CP: global constraints

Examples:

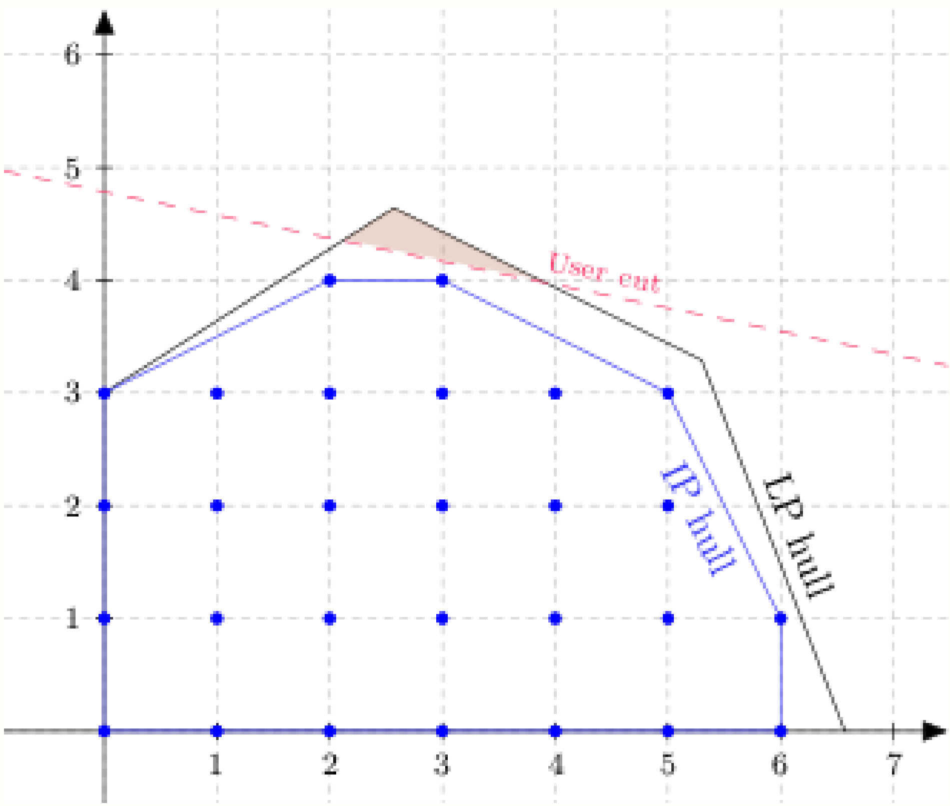
- alldifferent(X,Y,Z)
- *A*[X] = Y   with X,Y variables, A an array
- cumulative(...)   used in scheduling

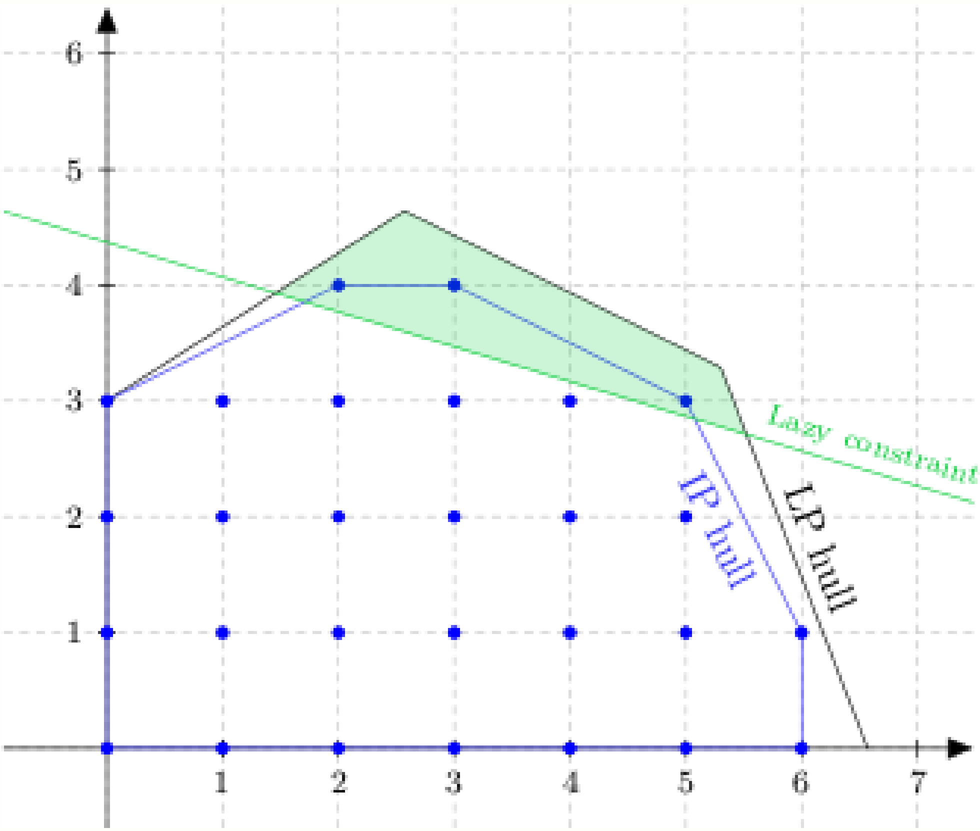
<u>model</u>: succinctly express a substructure

<u>solve</u>, with specialised algorithms:

- optimized data structures = more efficient
- (sometimes) more pruning = more effective

MIP: cutting planes and lazy cons





Source: orinanobworld.blogspot.com.au

How Are SAT/MIP/CP Solvers Being Used in ML/DM Settings?

Pure SAT/MIP/CP Formulation

Extensions To SAT/MIP/CP Solvers

Hybrid Formulations

Chaining of specialized algorithms with SAT/MIP/CP solvers

e.g. mine all patterns, then find a cover of those

Use SAT/MIP/CP solver as subproblem solver (e.g. as oracle)

e.g. generate candidate cluster, query feasability

How to choose between SAT/MIP/CP?

No free lunch!

General guidelines:

- if decision problem: try SAT first
- if inherently Boolean: try (max)SAT first
- if few constraints or natural to relax: try MIP first
- if highly complex constraints: try CP first

Pattern (Set) Mining

in Tabular Data

Overview

Itemset mining: Background, In CP, In SAT

Pattern set mining: Background, In CP, In MIP

Decision trees: Background, In SAT, MIP, CP

Overview

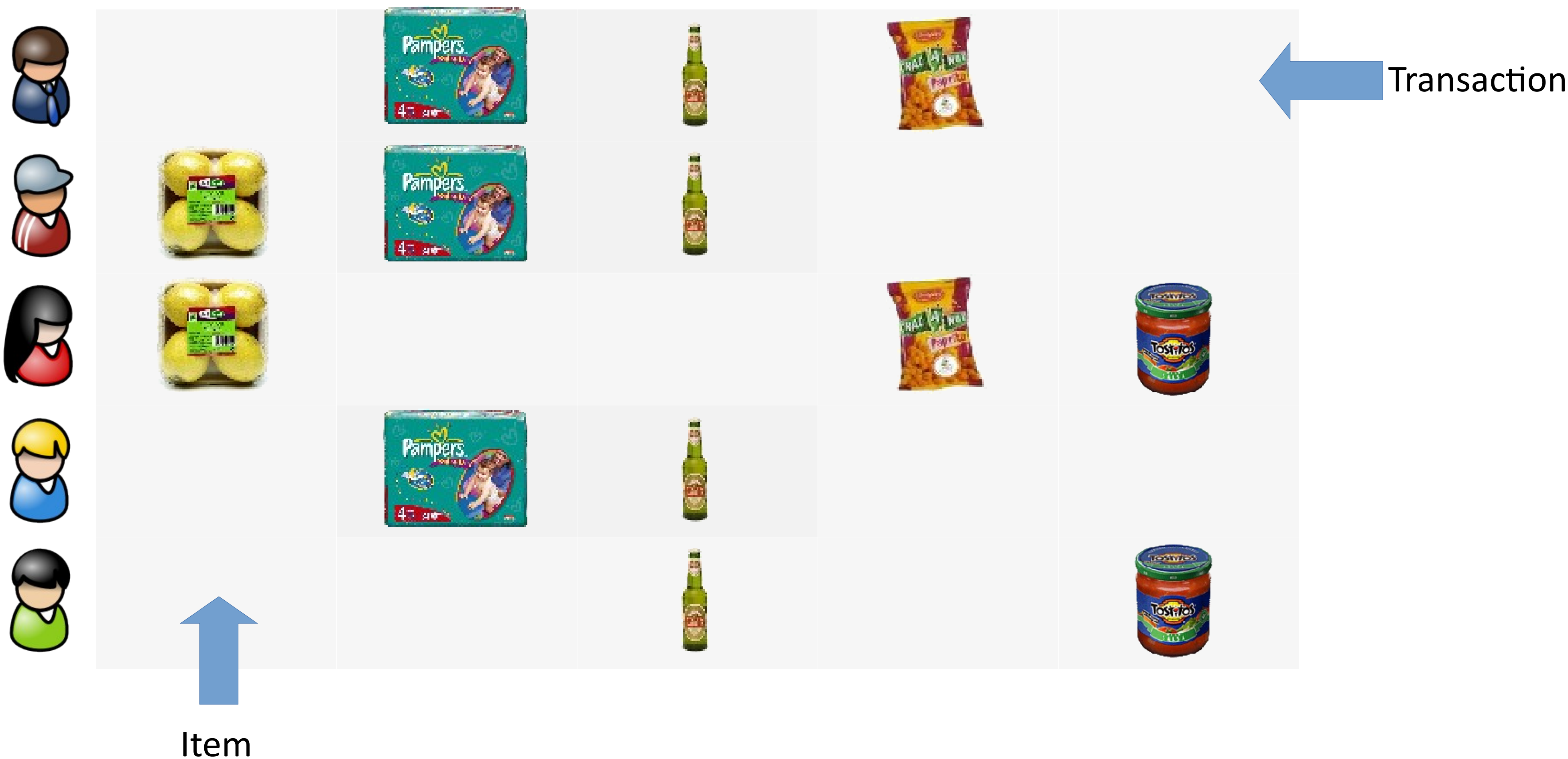
Itemset mining: **Background**, In CP, In SAT

Pattern set mining: Background, In CP, In MIP

Decision trees: Background, In SAT, MIP, CP

Frequent Itemset Mining

<u>1993</u>: The task of *frequent itemset mining* was defined by Agrawal, Imieliński, and Srikant for Boolean supermarket data



Frequent Itemset Mining

<u>Frequent itemset:</u> itemset that occurs in more than *minsup* transactions, where *minsup* is a user-defined threshold

Frequent Itemset Mining

Can be applied on any tabular dataset after binarization

A	B	C
0.5	0.6	10
0.3	0.7	12
0.9	0.4	9

A=[0-0.5]	A=]0.5-1]	B=[0-0.5]	B=]0.5-1]	C=[0-10]	C=]10-20]
1	0	0	1	1	0
1	0	0	1	0	1
0	1	1	0	1	0

Frequent Itemset Mining

Examples of data analyzed using FIM in the literature:

- Supermarket products per visitor
- Web pages accessed per visitor
- Binarized gene expression for patients
- Transcription factor binding sites of genes
- Single-nucleotide polymorphisms of patients

Frequent Itemset Mining

<u>Frequent itemset mining:</u> the problem of finding <u>all</u> frequent itemsets

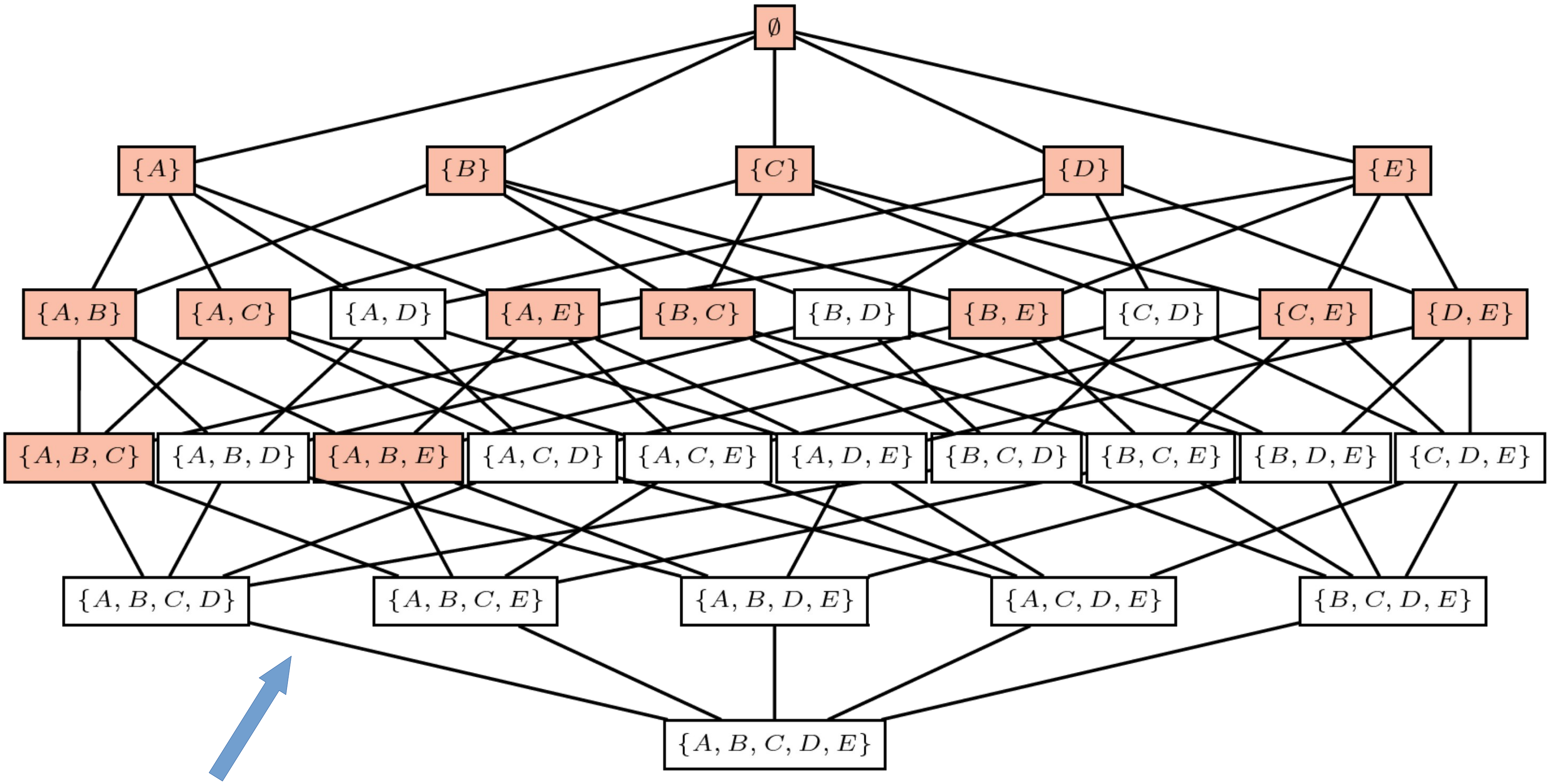
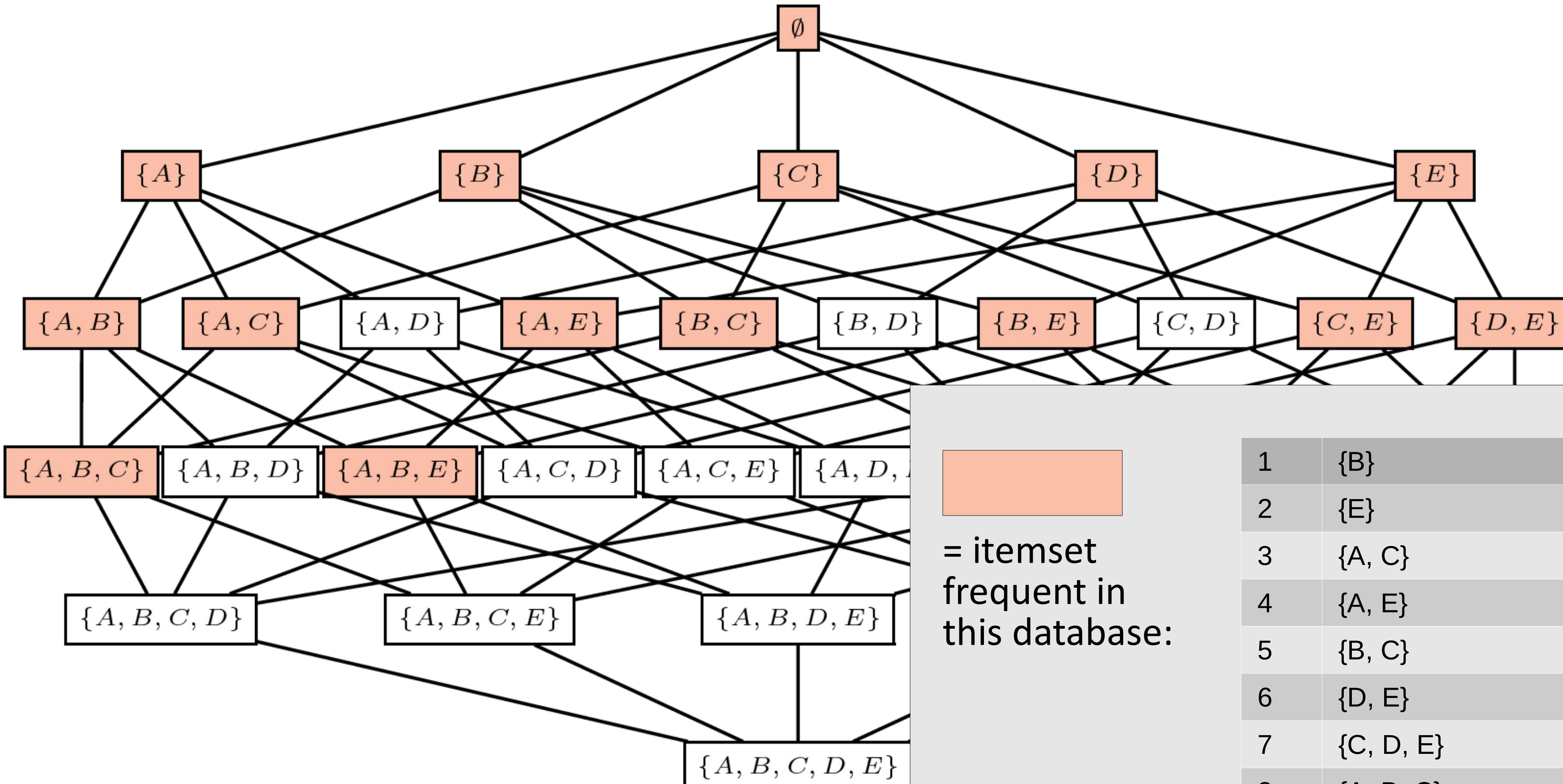


Diagram with all itemsets for a database with items {A,B,C,D,E}

Frequent Itemset Mining

Frequent itemset mining: the problem of finding all frequent itemsets



= itemset frequent in this database:

1	{B}
2	{E}
3	{A, C}
4	{A, E}
5	{B, C}
6	{D, E}
7	{C, D, E}
8	{A, B, C}
9	{A, B, E}
10	{A, B, C, E}

Frequent Itemset Mining

Challenge 1: how to find all frequent itemsets efficiently?

Frequent Itemset Mining

1994-2004: Development of efficient frequent itemset mining algorithms

- Many FIM implementations available at http://fimi.ua.ac.be .
- Also implementation choices are important for performance
- Small improvements not accepted at data mining conferences any more

Itemset Mining

<u>Challenge 1:</u> how to find all frequent itemsets efficiently?

<u>Challenge 2</u>: how to determine small sets of useful patterns?

Itemset Mining

Challenge 1: how to find all frequent itemsets efficiently?

Challenge 2: how to determine small sets of useful patterns?

- Constraint-based data mining
- Pattern set mining

Itemset Mining

Challenge 1: how to find all frequent itemsets efficiently?

Challenge 2: how to determine small sets of useful patterns?

- **Constraint-based data mining**
- Pattern set mining

Itemset Mining

Challenge 1: how to find all frequent itemsets efficiently?

Challenge 2: how to determine small sets of useful patterns?

- **Constraint-based data mining**
  - Constraints based on data
  - Constraints based on relationships between patterns
  - Constraints based on syntax
- Pattern set mining

Itemset Mining

Challenge 1: how to find all frequent itemsets efficiently?

Challenge 2: how to determine small sets of useful patterns?

- Constraint-based data mining
  - **Constraints based on data**
  - Constraints based on relationships between patterns
  - Constraints based on syntax
- Pattern set mining

Itemset Mining

Challenge 1: how to find all frequent itemsets efficiently?

Challenge 2: how to determine small sets of useful patterns?

- Constraint-based data mining
  - **Constraints based on data**
  - Constraints based on relationships between patterns
  - Constraints based on syntax
- Pattern set mining

Itemset Mining

Challenge 1: how to find all frequent itemsets efficiently?

Challenge 2: how to determine small sets of useful patterns?

- Constraint-based data mining
  - Constraints based on data
    Minimum support in unsupervised data
  - Constraints based on relationships between patterns
  - Constraints based on syntax
- Pattern set mining

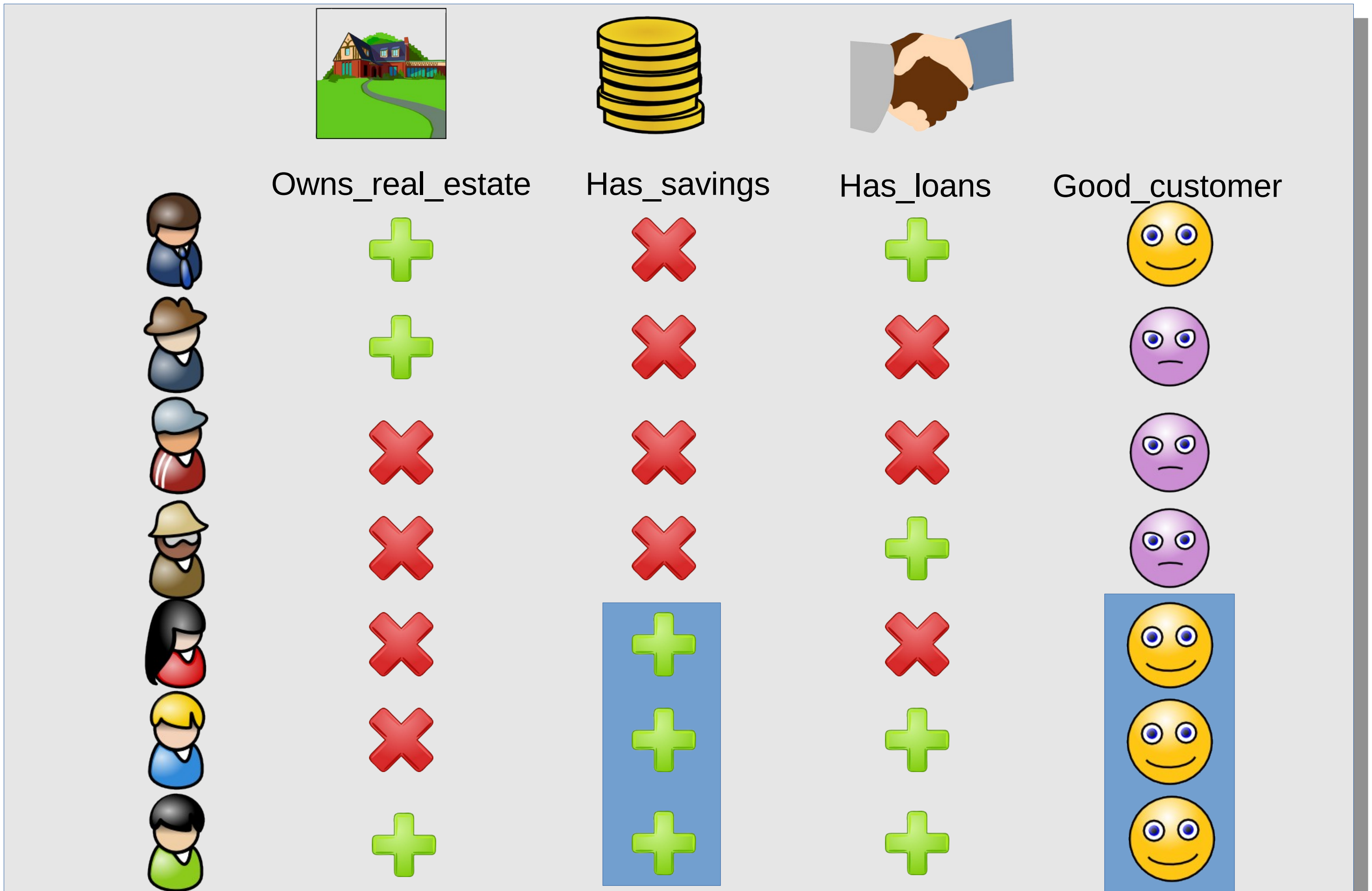
Itemset Mining

<u>Challenge 1:</u> how to find all frequent itemsets efficiently?

<u>Challenge 2</u>: how to determine small sets of useful patterns?

- Constraint-based data mining
  - Constraints based on data
    - Minimum support in unsupervised data
    - **<u>Supervised data</u>**
  - Constraints based on relationships between patterns
  - Constraints based on syntax
- Pattern set mining

Pattern Constraints on Data: Supervised Data



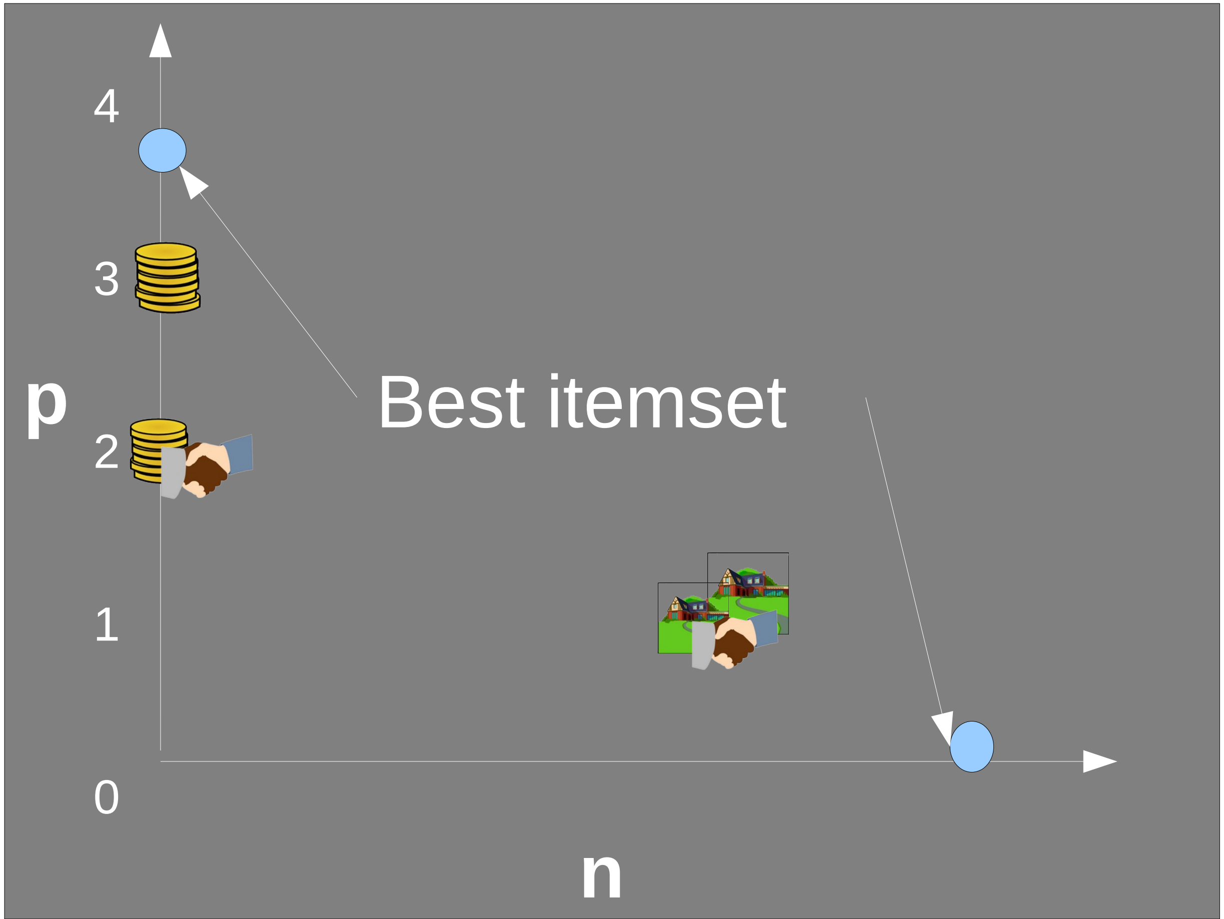
Contingency Table

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

Constraints in Supervised Data: PN-space

Contingency Table

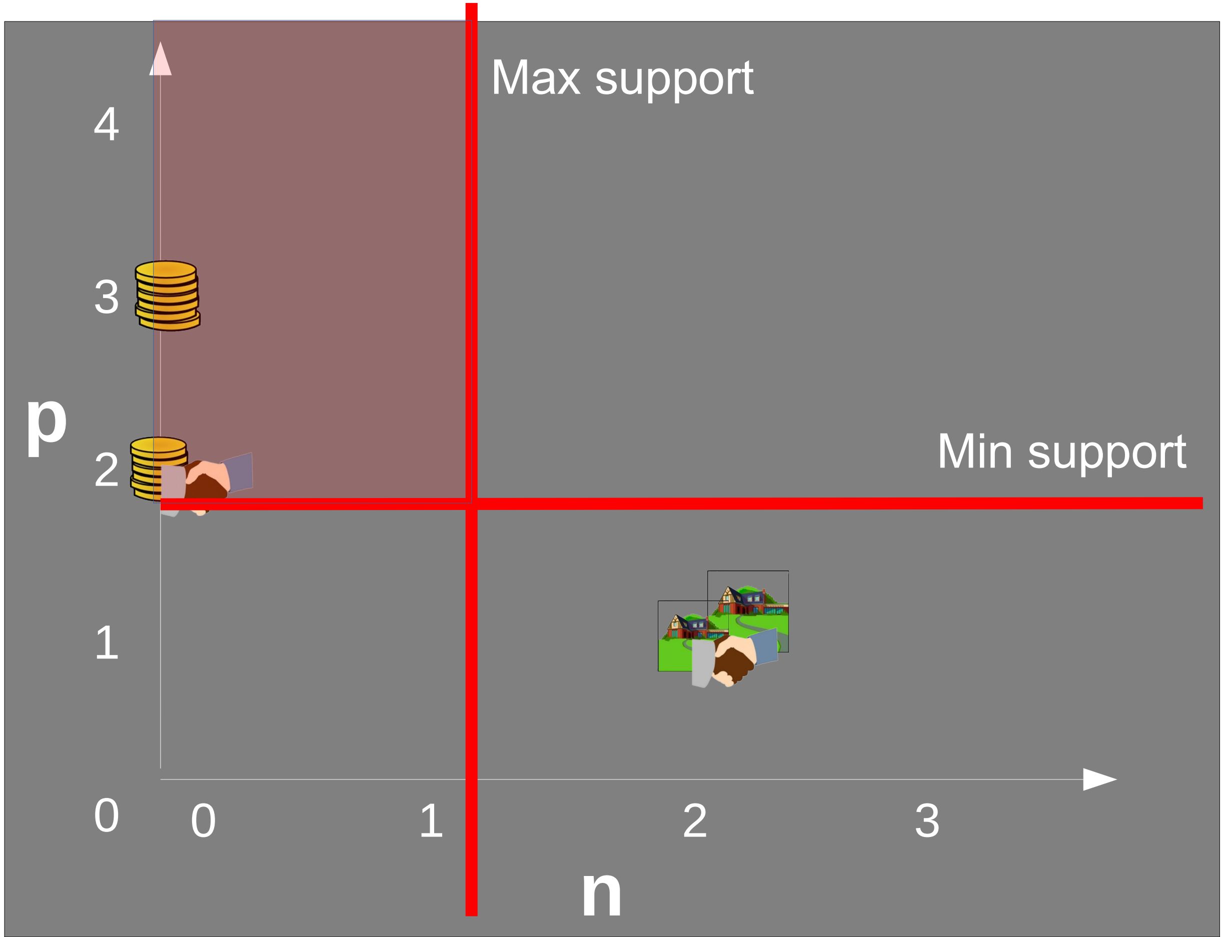
TP: 3 (=p)	FP: 0 (=n)	3
FN: 1	TN: 3	4
P: 4	N: 3	



Constraints in Supervised Data: Min & Max Support

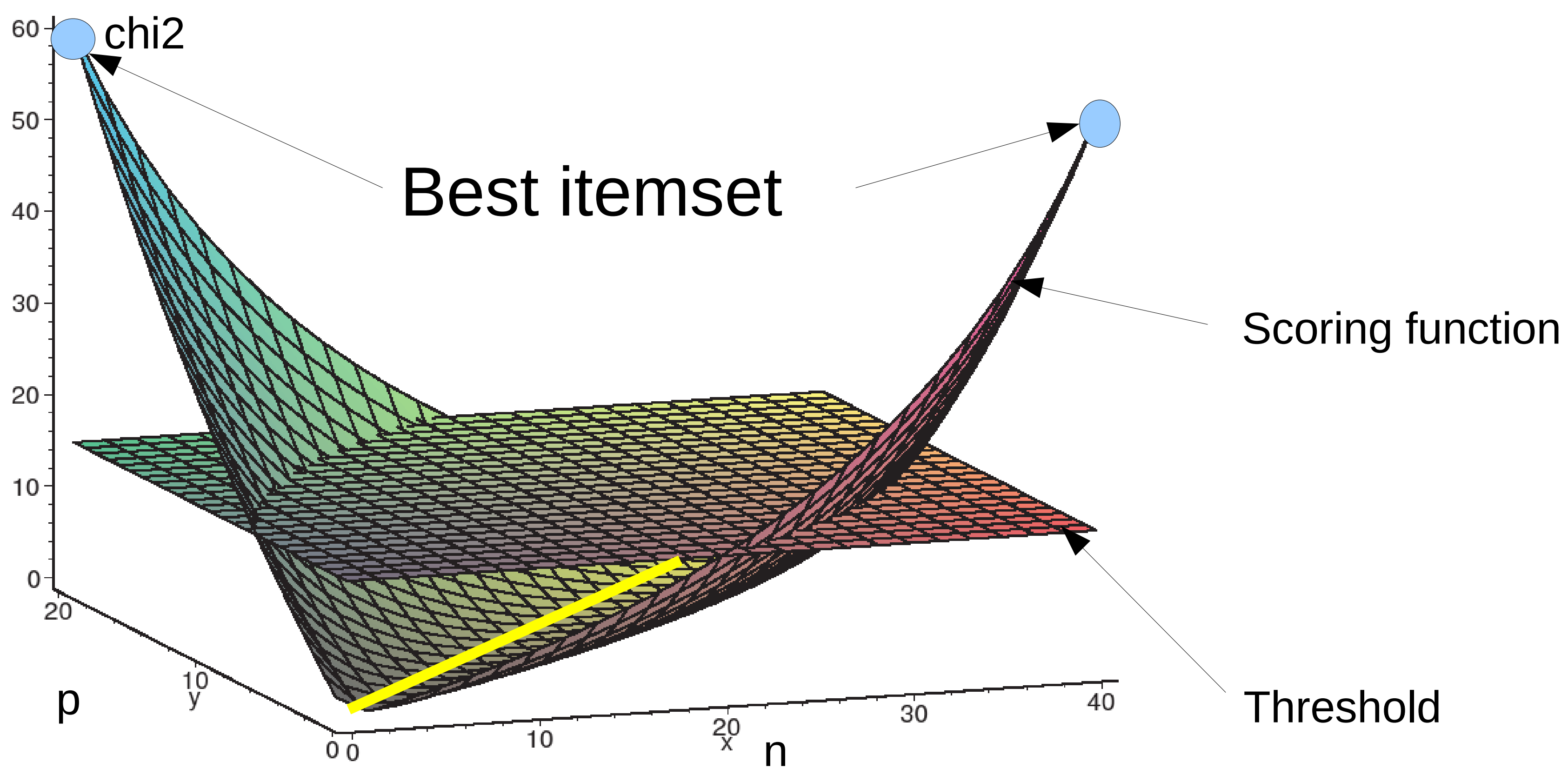
Contingency Table

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



(Reverse Min/ Max support for the other corner)

Constraints in Supervised Data: Threshold on Scoring Function



How do we call such Patterns?

Itemset Mining

Challenge 1: how to find all frequent itemsets efficiently?

Challenge 2: how to determine small sets of useful patterns?

- Constraint-based data mining
  - Constraints based on data
  - **Constraints based on relationships between patterns**
  - Constraints based on syntax
- Pattern set mining

Itemset Mining

Challenge 1: how to find all frequent itemsets efficiently?

Challenge 2: how to determine small sets of useful patterns?

- Constraint-based data mining
  - Constraints based on data
  - **Constraints based on relationships between patterns**
  - Constraints based on syntax
- Pattern set mining

Itemset Mining

Challenge 1: how to find all frequent itemsets efficiently?

Challenge 2: how to determine small sets of useful patterns?

- Constraint-based data mining
  - Constraints based on data
  - **Constraints based on relationships between patterns**
    Closed patterns
    Maximally frequent patterns
  - Constraints based on syntax
- Pattern set mining

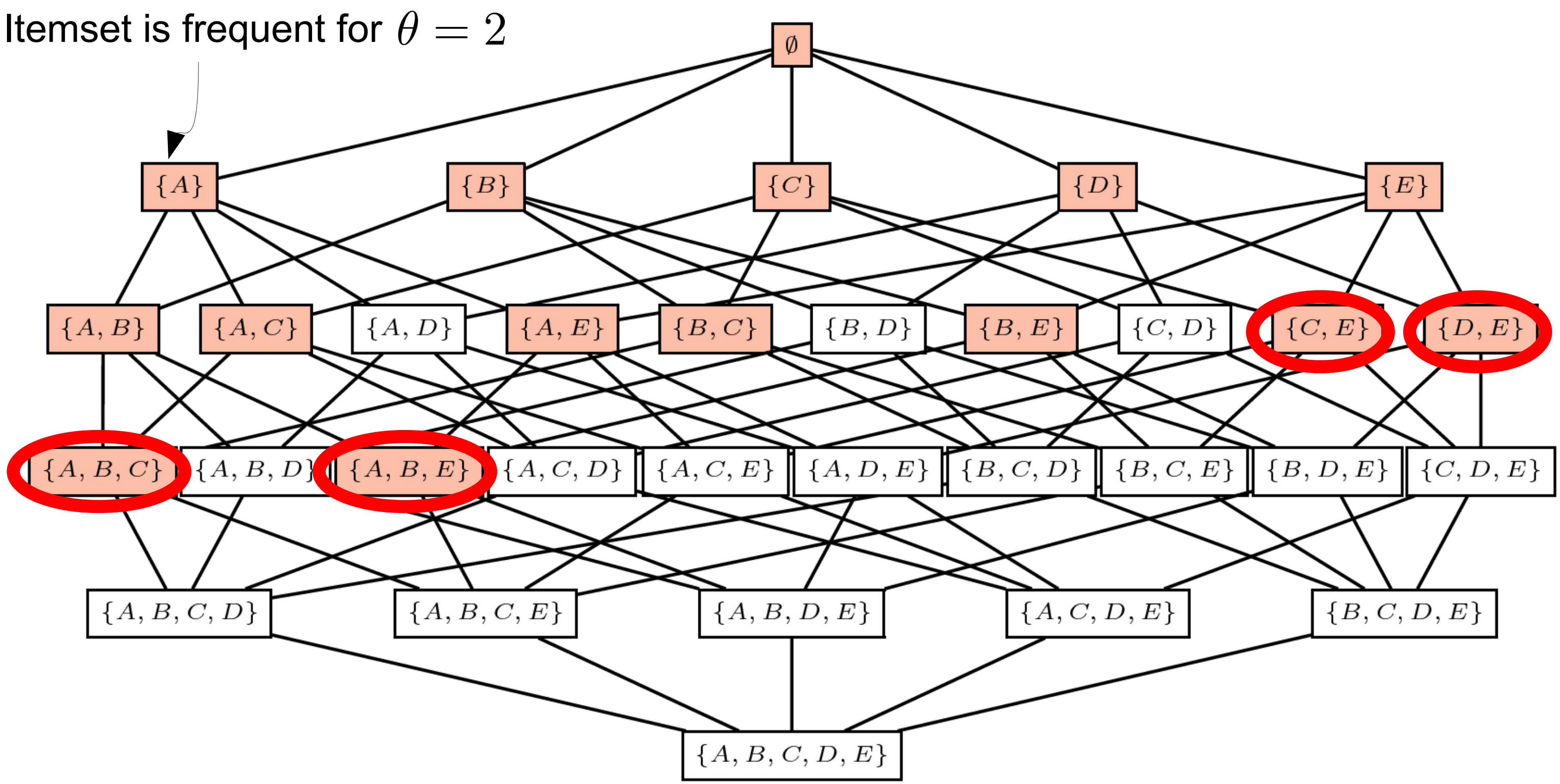
Itemset Mining

Challenge 1: how to find all frequent itemsets efficiently?

Challenge 2: how to determine small sets of useful patterns?

- Constraint-based data mining
  - Constraints based on data
  - Constraints based on relationships between patterns
    Closed patterns
    Maximally frequent patterns
  - Constraints based on syntax
- Pattern set mining

Maximal Frequent Itemsets



Maximal = One cannot add an item and still have a frequent itemset

Itemset Mining

Challenge 1: how to find all frequent itemsets efficiently?

Challenge 2: how to determine small sets of useful patterns?

- Constraint-based data mining
  - Constraints based on data
  - Constraints based on relationships between patterns
  - **Constraints based on syntax**
- Pattern set mining

C) Constraints based on Syntax

On example: suppose every item has a cost *c(i)*

Find itemsets under such constraints:

- Max cost: $\sum_{i \in I} c(i) \leq \theta$
- Min cost: $\sum_{i \in I} c(i) \geq \theta$
- Max average cost: $\sum_{i \in I} c(i)/|I| \leq \theta$
- Min average cost: $\sum_{i \in I} c(i)/|I| \geq \theta$

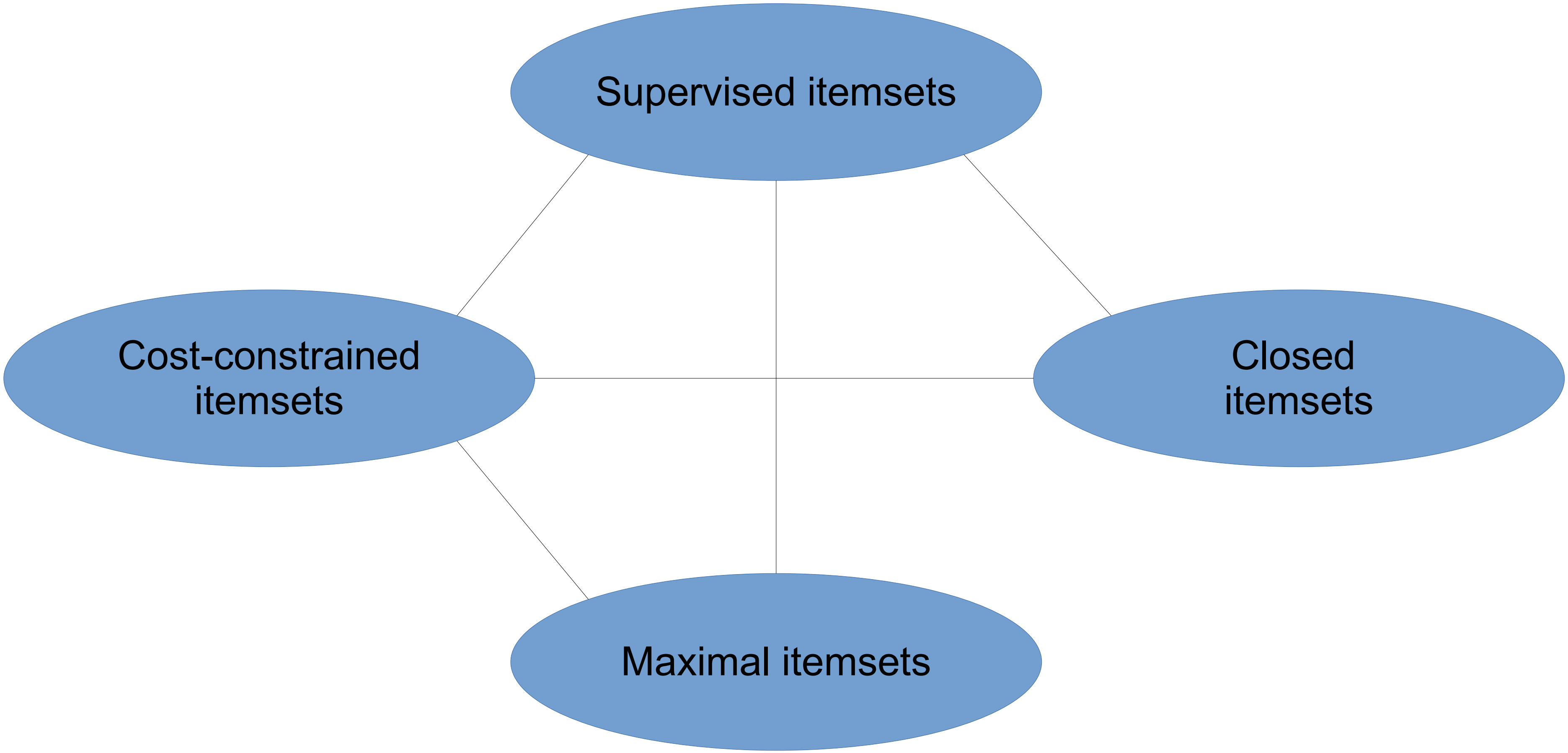
In the Literature

Classic Setting

Algorithms for many settings:

- Closed itemset mining
  LCM, modifications of ECLAT, FP-Growth, …
- Maximal frequent itemset mining
  MaxMiner, Mafia, modifications of ECLAT, FP-Growth, …
- Supervised itemset mining
  DDPMine, Cortana, Opus, …
- Cost-constrained itemset mining
  FP-Bonsai, Exante, DualMiner, ...

Combinations of Constraints



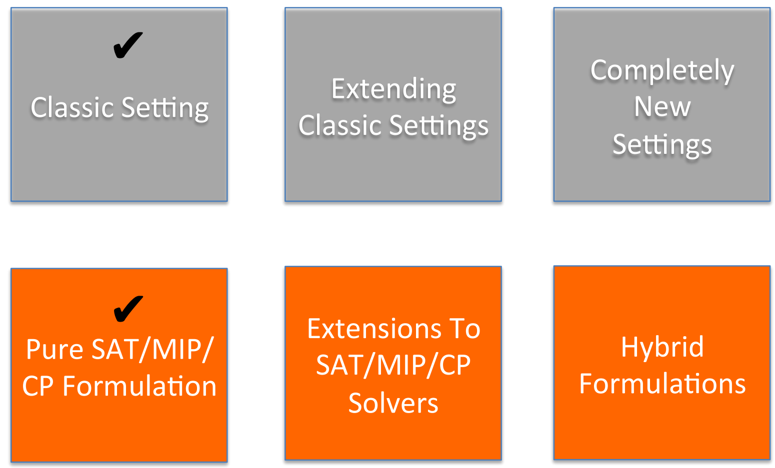
Overview

Itemset mining: Background, **In CP**, In SAT

Pattern set mining: Background, In CP, In MIP

Decision trees: Background, In SAT, MIP, CP

Itemset Mining
using Pure CP Formulations



Can we use CP systems for solving constraint-based itemset mining problems?

L. De Raedt, T. Guns, S. Nijssen. Constraint programming for itemset mining. KDD 2008:

$\rightarrow$ YES!

Initial Models

Frequent itemset mining,
closed itemset mining,
maximal frequent itemset mining,
cost-based constraints

Classic Setting

Solved using the generic CP system *Gecode* to enumerate all solutions

Pure CP Formulation

Modeling Frequent Itemset Mining

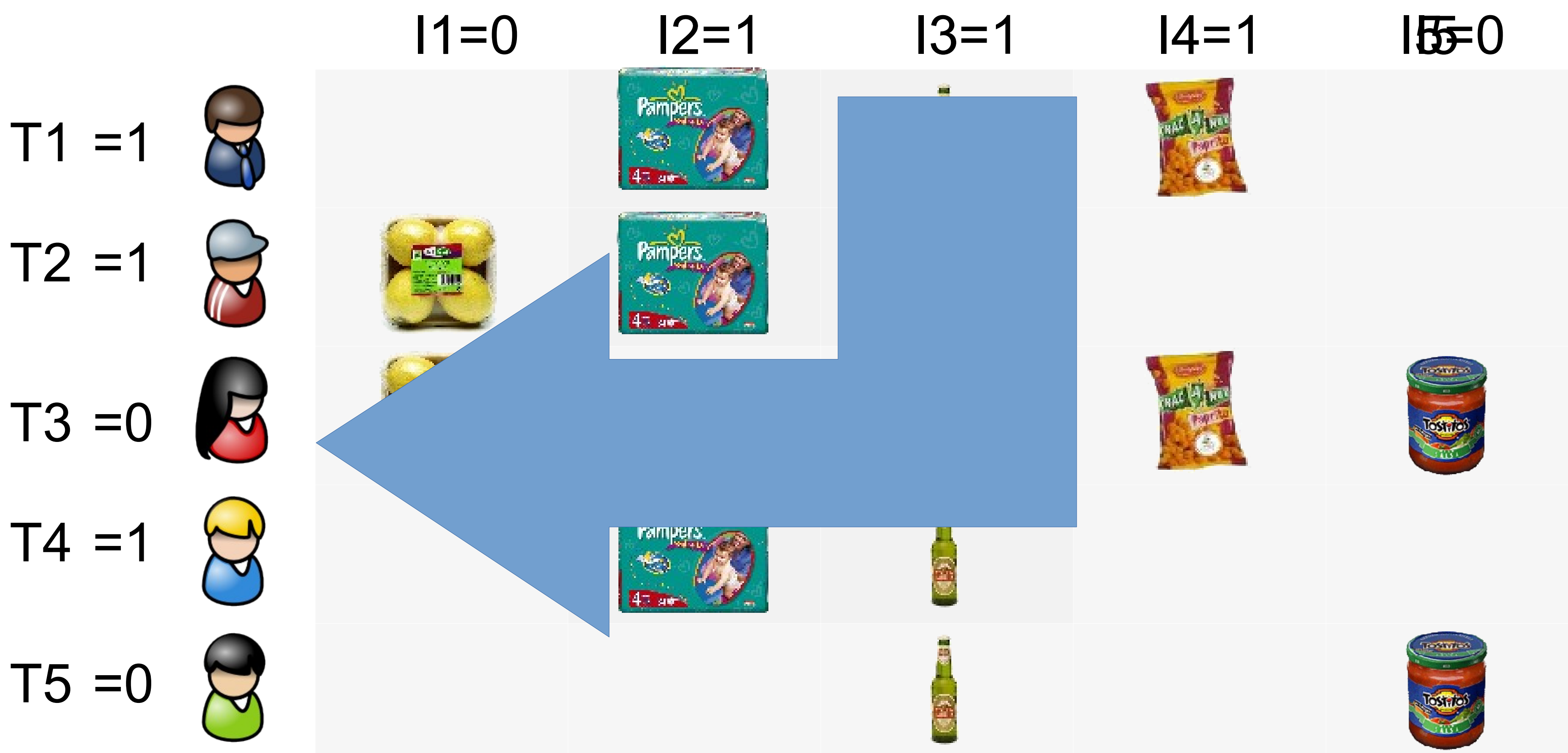
One Boolean variable per item

One Boolean variable per transaction

Modeling Frequent Itemset Mining

Two constraints:

1) A <u>coverage constraint</u>: (Tj = 1) iff (Itemset in Tj)



Modeling Frequent Itemset Mining

Two constraints:

2) A <u>support constraint</u>: $\sum_j Tj \geq minsup$

I1=0 I2=1 I3=1 I4=1 I5=0

T1 =1

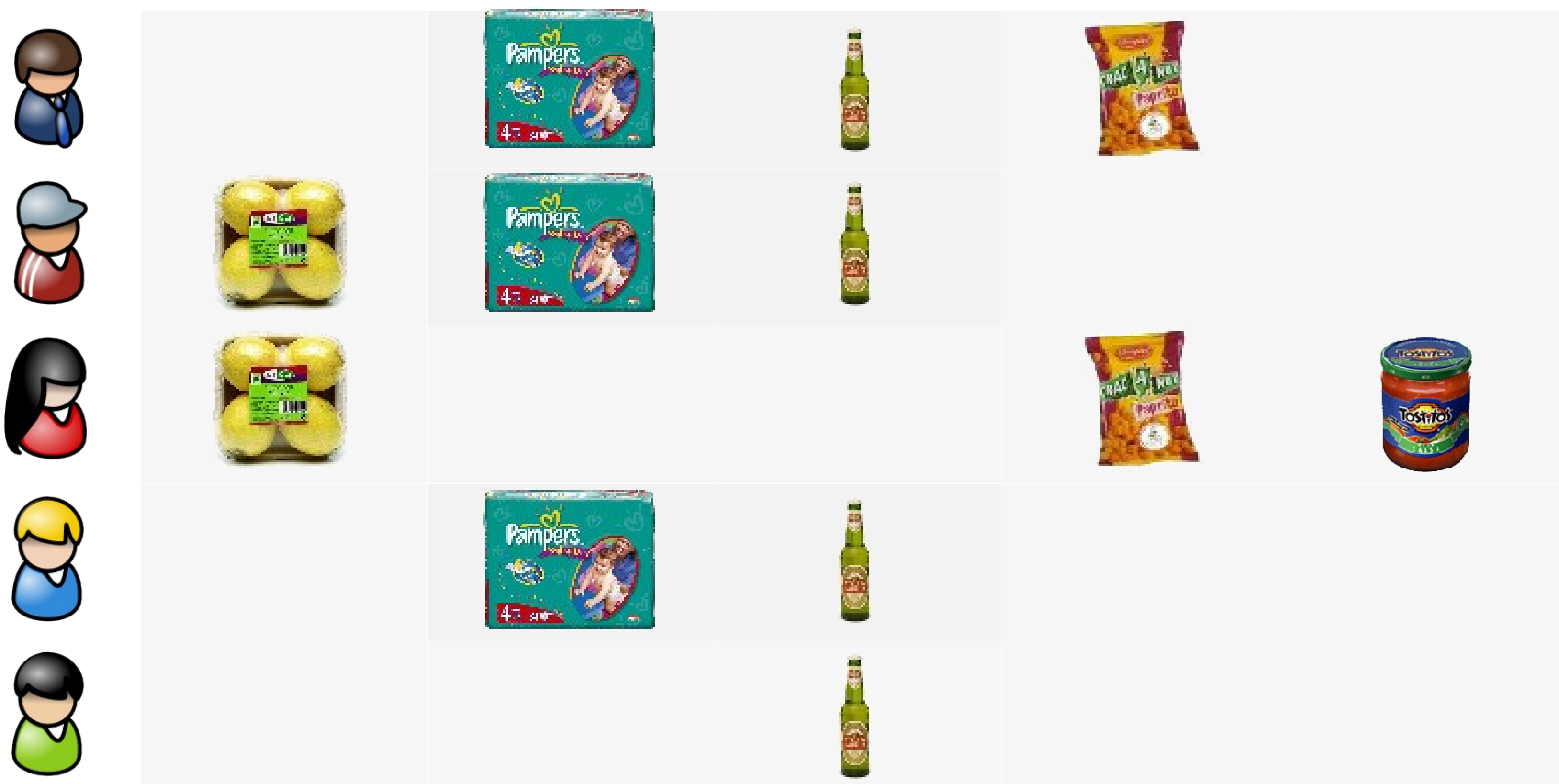
T2 =1

T3 =0

T4 =1

T5 =0

3



Modeling Frequent Itemset Mining: Implementation Tweaks

Gecode solver with existing constraints, not changing the solver

Pure CP Formulation

The resulting model is not easy to read:

$$I_t = 1 \rightarrow \sum_{t \in \mathcal{T}} T_t I_i \geq \theta$$ (Support)

$$T_t = 1 \Leftrightarrow \sum_{i \in \mathcal{I}} (1 - \mathcal{D}_{ti}) I_i = 0$$ (Coverage)

Two “reified summation constraints”

Other Itemset Mining Tasks

Pure CP Formulation

Similar results for:

- Closed itemset mining → LCM

$$I_t = 1 \rightarrow \sum_{t \in \mathcal{T}} T_t I_i \geq \theta$$

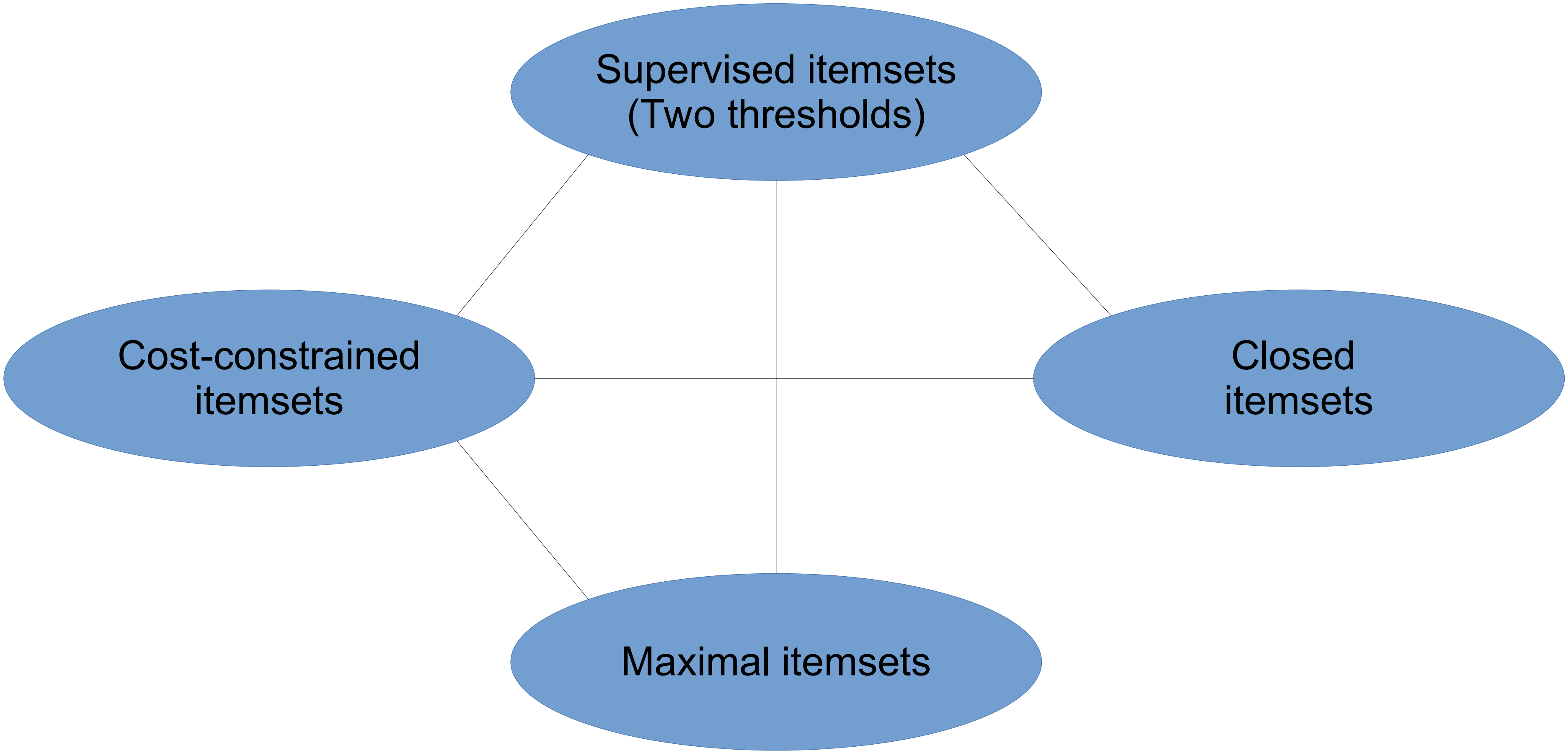
$$T_t = 1 \Leftrightarrow \sum_{i \in \mathcal{I}} (1 - \mathcal{D}_{ti}) I_i = 0$$

$$I_i = 1 \Leftrightarrow \sum_{t \in \mathcal{T}} (1 - \mathcal{D}_{ti}) T_t = 0$$

- Maximal frequent itemset mining → MAFIA
- Cost-based itemset mining → Exante

Combinations of Constraints

Extending Classic Settings



Easy in CP!

Constraints in Supervised Data

Siegfried Nijssen, Tias Guns, Luc De Raedt. Correlated itemset mining in ROC space. KDD 2009.

For supervised patterns we added a constraint to Gecode

Extensions To CP

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+*

cimcp = our implementation in Gecode

ddpmine = algorithm from the literature

lcm = post-processing frequent itemsets

corrmine = implementation of all propagation in more specialized algorithm

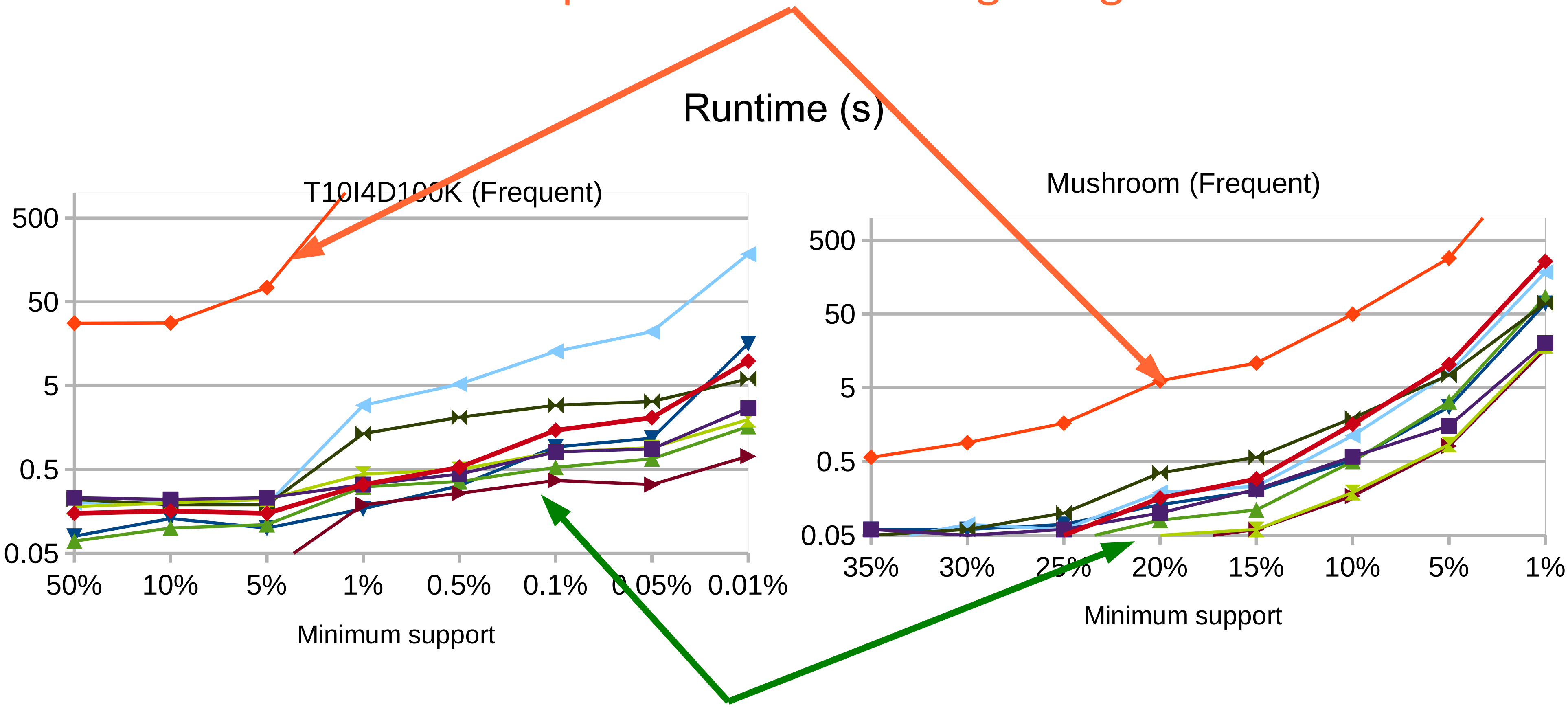
\- = algorithm crashes

> = time-out

Some Results

Two Datasets from the FIMI challenge

Frequent itemset mining using Gecode



Frequent itemset mining using algorithms from the FIMI challenge

Itemset Mining using Extended CP Systems

✔ Classic Setting	Extending Classic Settings	Completely New Settings
Pure SAT/MIP/ CP Formulation	✔ Extensions To SAT/MIP/CP Solvers	Hybrid Formulations

Is it possible to build or modify CP systems that execute itemset mining tasks more efficiently?

A New CP System for More Efficient Itemset Mining

Siegfried Nijssen, Tias Guns. Integrating constraint programming and itemset mining. ECML PKDD 2010.

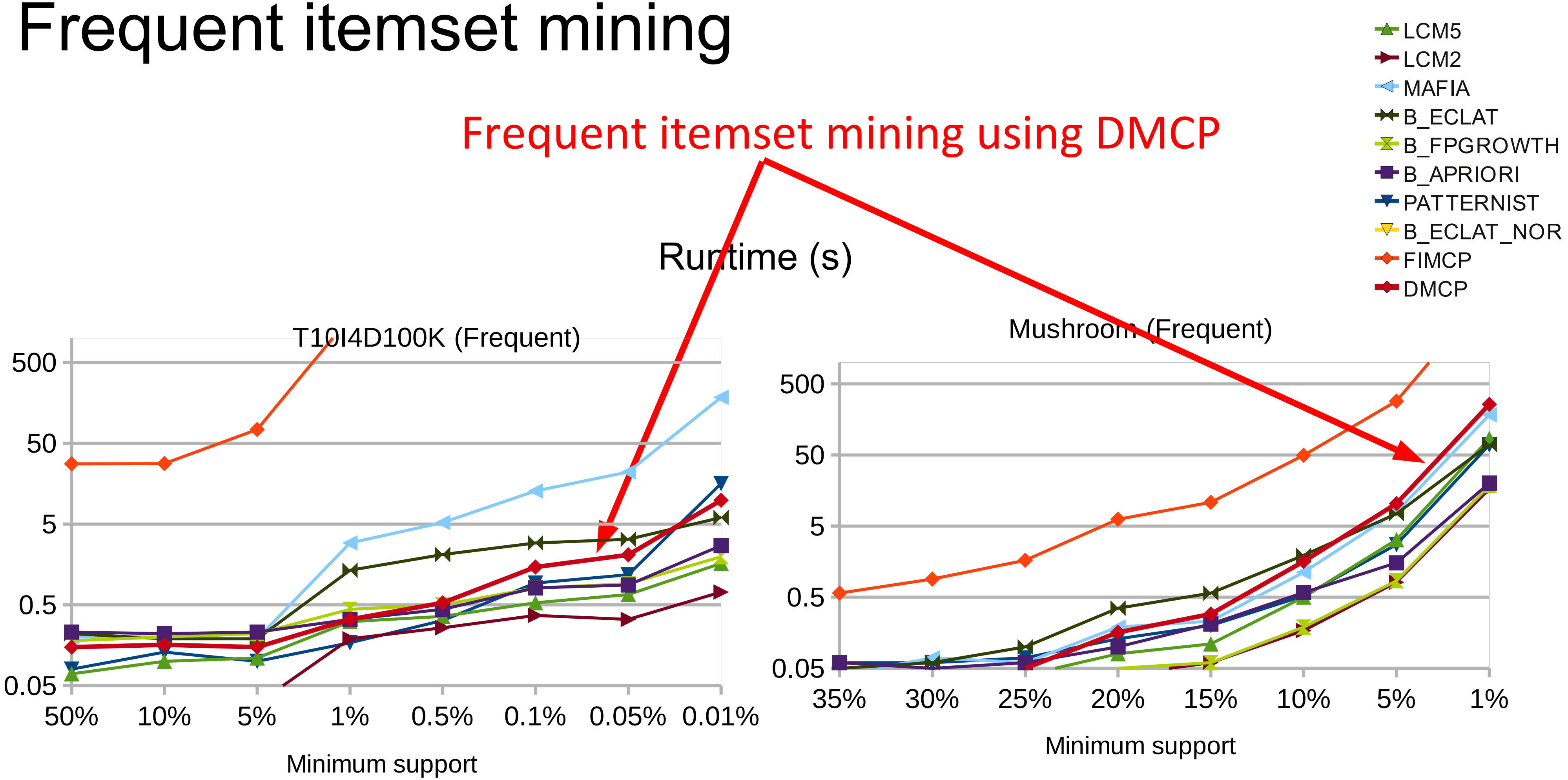
Is it possible to build a **new** CP system that executes itemset mining tasks more efficiently?

DMCP: a CP system that only supports

- Boolean vector variables
- Binary constraints between these Boolean vectors

DMCP: Some Results

Frequent itemset mining



Adding Global Constraints to CP Systems for Itemset Mining

Can we add new global constraints to **existing** CP systems, such that their performance on itemset mining tasks improves?

Extensions To CP

- ClosedPattern global constraint, integrated in the Gecode solver

M. Maamar, N. Lazaar, S. Loudni, Y. Lebbah. A global constraint for closed itemset mining. CP 2016.

- CoverSize and a CoverClosure global constraint, integrated in the Oscar solver

P. Schaus, J. Aoga, T. Guns. CoverSize: A global constraint for frequence-based itemset mining. CP 2017.

- FrequentSubs and InfrequentSupers global constraints, integrated in the Oscar solver

M. Belaid, C. Bessiere, N. Lazaar. Constraint Programming for Mining Borders of Frequent Itemsets. IJCAI 2019.

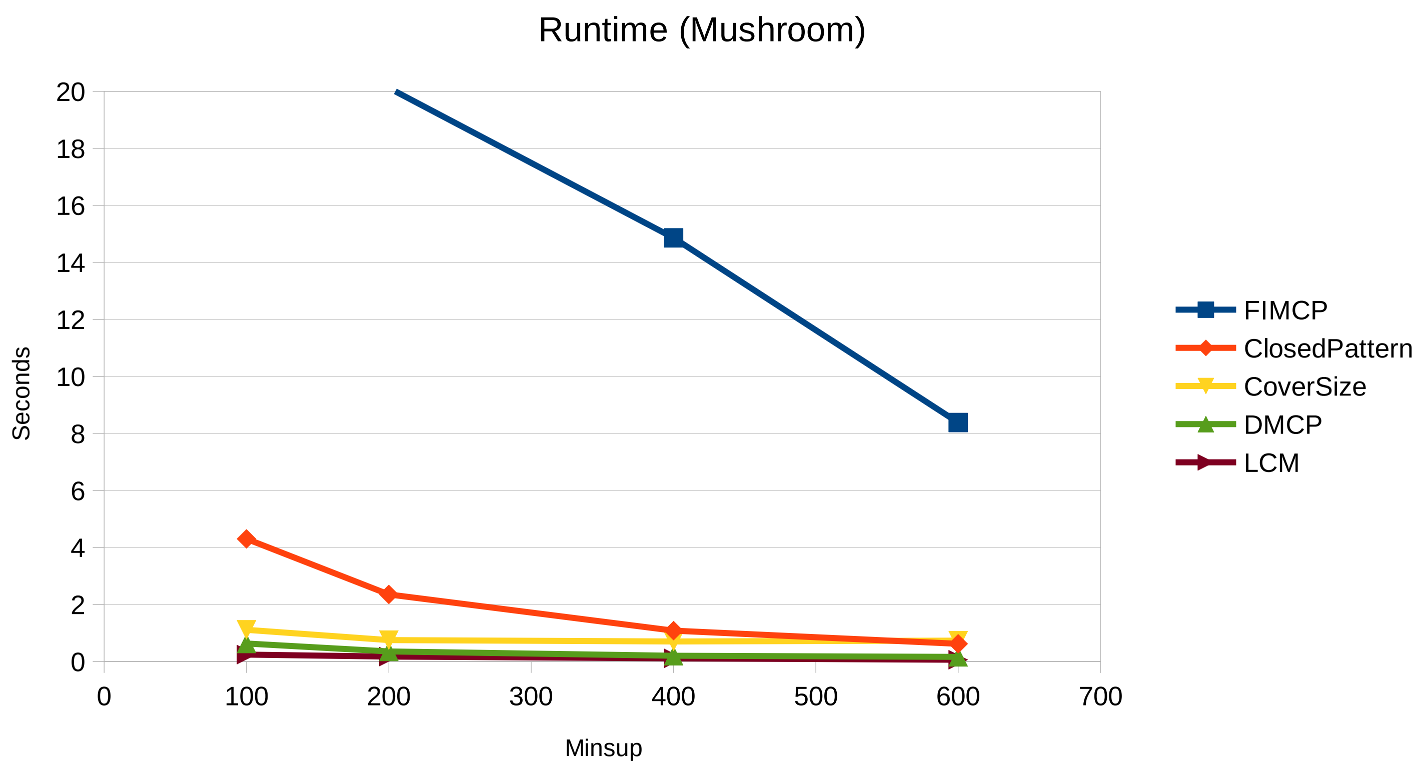
Theoretical Comparison

	FIMCP	DMCP	Closed Pattern	CoverSize/ Cover Closure	Infrequent Supers
Variables for items	X	X	X	X	X
Variables for transactions	X	X	-	-	-
Variable for support	X	-	-	X	X
Closed patterns	X	X	X	X	-
Maximal patterns	X	X	-	-	X

Variables need to be present to allow for the addition of constraints
Exposing variables may degrade performance

Practical Comparison

One illustrative example for closed itemset mining



Overview

Itemset mining: Background, In CP, **In SAT**

Pattern set mining: Background, In CP, In MIP

Decision trees: Background, In SAT, MIP, CP

Itemset Mining using SAT Solvers

✔ Classic Setting	Extending Classic Settings	Completely New Settings
✔ Pure SAT/MIP/CP Formulation	Extensions To SAT/MIP/CP Solvers	Hybrid Formulations

Given progress in SAT solvers, can SAT solvers solve itemset mining problems (more) efficiently?

Modeling Itemset Mining in CNF

R. Henriques, I. Lynce, V. Manquinho. On When and How to use SAT to Mine Frequent Itemsets. ArXiv:1207.6253. 2012.

S. Jabbour, L. Sais, Y. Salhi. The top-k frequent closed itemset mining using top-k sat problem. ECML PKDD 2013.

Coverage can be modeled in a similar fashion as in CP systems

Example: coverage for transaction *T* with item *C* in database with items A,B and C is encoded as:

$$((\neg T) \rightarrow (\neg I_A \vee \neg I_B)) \wedge ((\neg I_A) \rightarrow \neg T) \wedge ((\neg I_B) \rightarrow \neg T)$$

Support is more complex; it requires the use of an encoding using sorting networks, cardinality networks, sequential counters…

However, these encodings are well-studied in the literature

Modeling Itemset Mining in CNF

Closed itemset mining & maximal frequent itemset mining can de moedeled similarly

Main challenge: we need to enumerate all solutions (AllSAT)

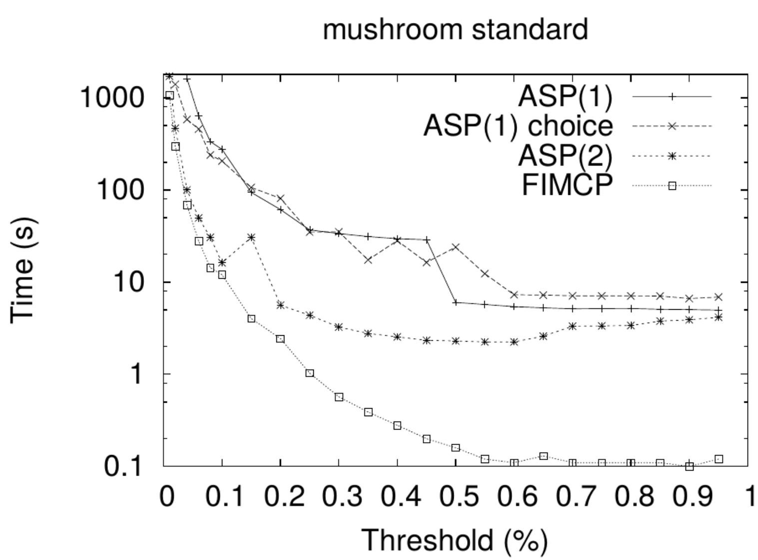
Compared to CP-based approaches, performance not competitive

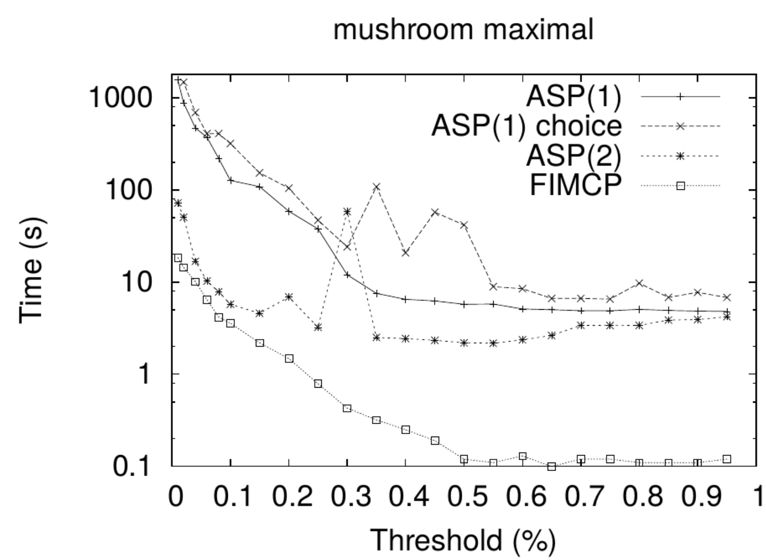
Itemset Mining using Answer Set Programming

M. Järvisolo. Itemset Set Mining as a Challenge for Anser Set Enumeration. LPNMR 2011.

ASP provides better support for solution enumeration

```
item(I) :- db( ,I).
transaction(T) :- db(T, ).
{ in itemset(I) } :- item(I).
in_support(T) :- { conflict at(T,I) : item(I) } 0, transaction(T).
conflict at(T,I) :- not db(T,I), in itemset(I), transaction(T).
:- { in support(T) } N-1, threshold(N).
```





Overview

Itemset mining: Background, In CP, In SAT

Pattern set mining: **Background**, In CP, In MIP

Decision trees: Background, In SAT, MIP, CP

Pattern Set Mining

Problem of constraint-based pattern mining: patterns often are still similar to each other

A	B	C	D	E	F	G	H
1	1	1	1	0	0	0	1
1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	1	1	1	1
1	1	1	0	0	0	0	1

Closed patterns
under support constraint 2:

A B C
A B C D
A B C D E F G
A B C H
B C D
B C D E F G
B C D H
B C
B C F
B C H

Pattern Set Mining

Problem of constraint-based pattern mining: patterns often are still similar to each other

A	B	C	D	E	F	G	H
1	1	1	1	0	0	0	1
1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	1	1	1	1
1	1	1	0	0	0	0	1

Closed patterns
under support constraint 2:

A B C
A B C D
A B C D E F G
A B C H
B C D
B C D E F G
B C D H
B C
B C F
B C H

Pattern Set Mining

Problem of constraint-based pattern mining:
patterns often are still similar to each other

A	B	C	D	E	F	G	H
1	1	1	1	0	0	0	1
1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	1	1	1	1
1	1	1	0	0	0	0	1

Closed patterns
under support constraint 2:

A B C
A B C D
A B C D E F G
A B C H
B C D
B C D E F G
B C D H
B C
B C F
B C H

Pattern Set Mining

Pattern Set Mining aims to take into account the aim of the data miner when selecting subsets of patterns

A	B	C	D	E	F	G	H
1	1	1	1	0	0	0	1
1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	1	1	1	1
1	1	1	0	0	0	0	1

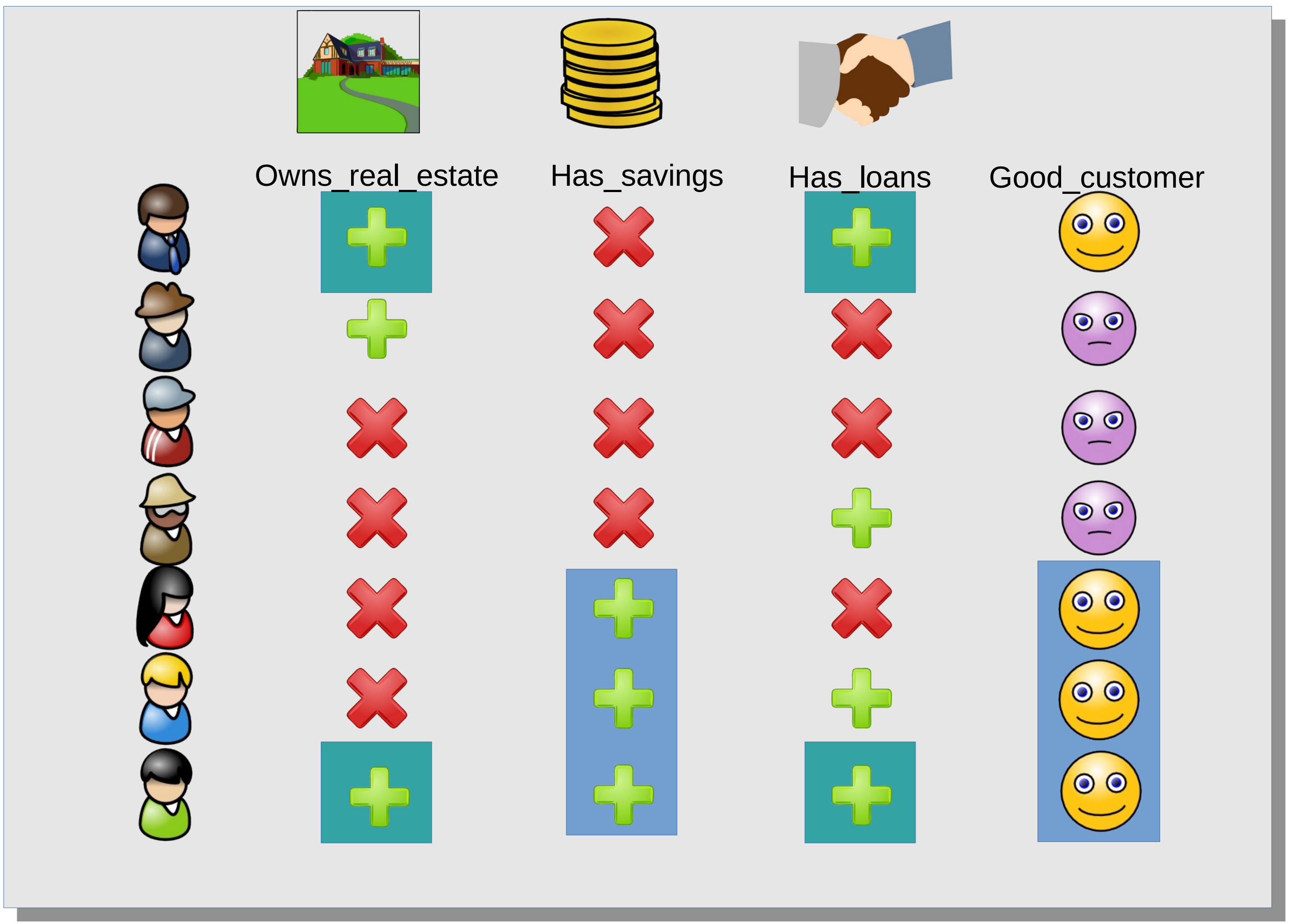
If her aim is to obtain a good summary of the data this set of closed itemsets may be much more better:

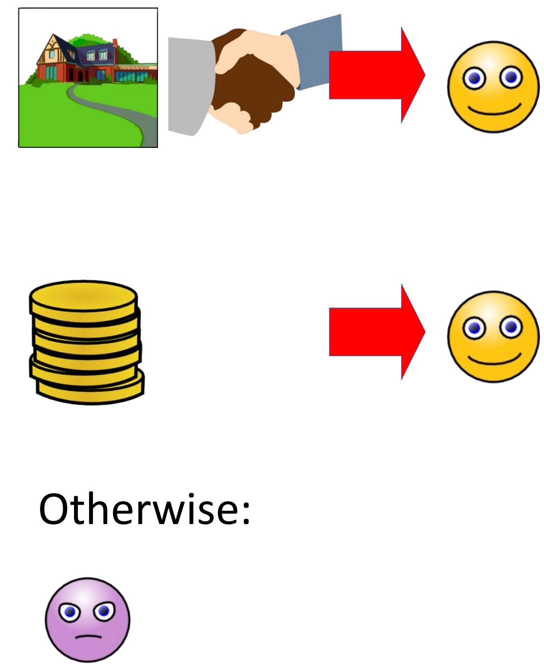
A B C
B C D E F G

This set is a tiling that covers most 1s with only two patterns

Pattern Set Mining

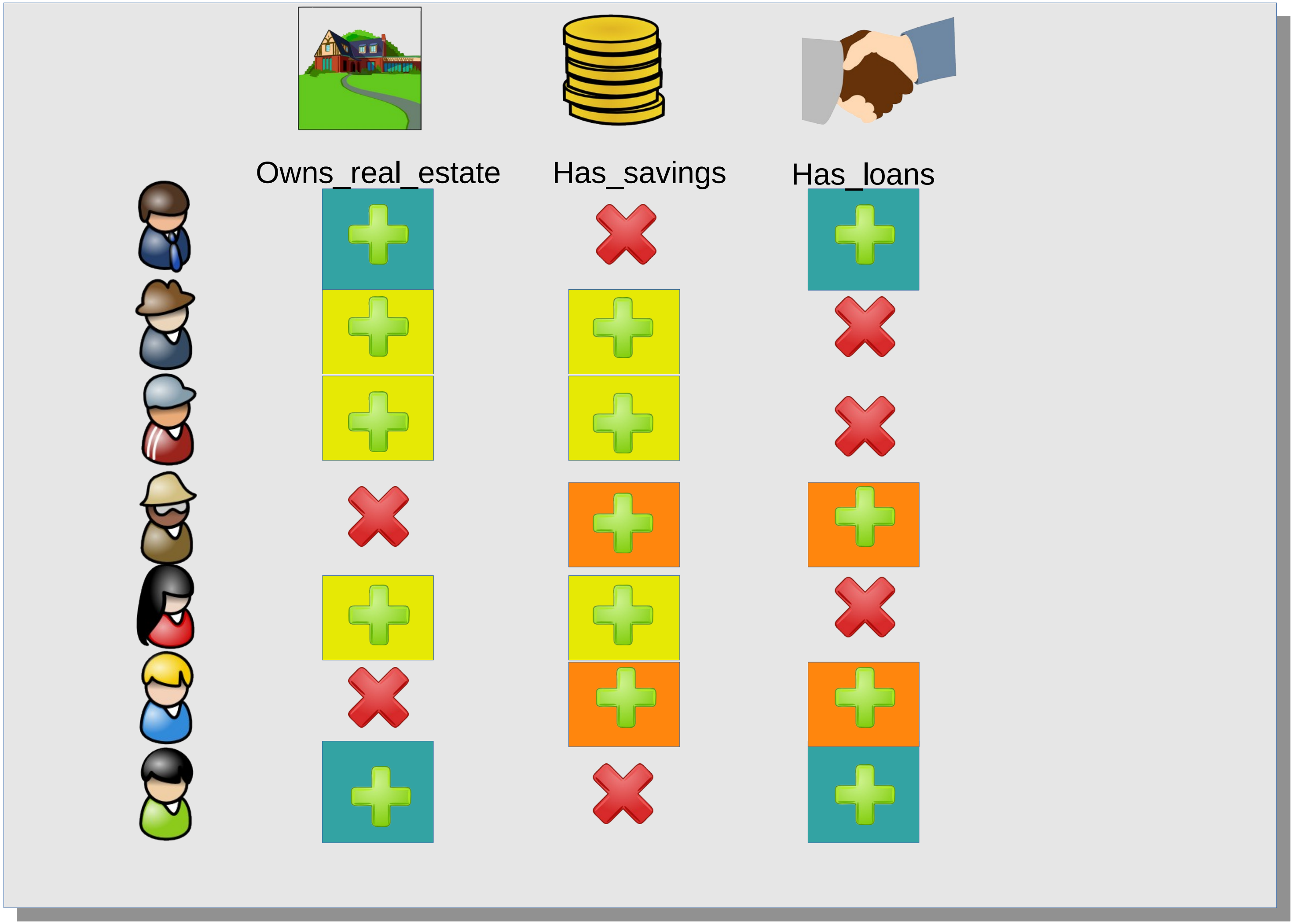
In supervised settings = rule learning/concept learning

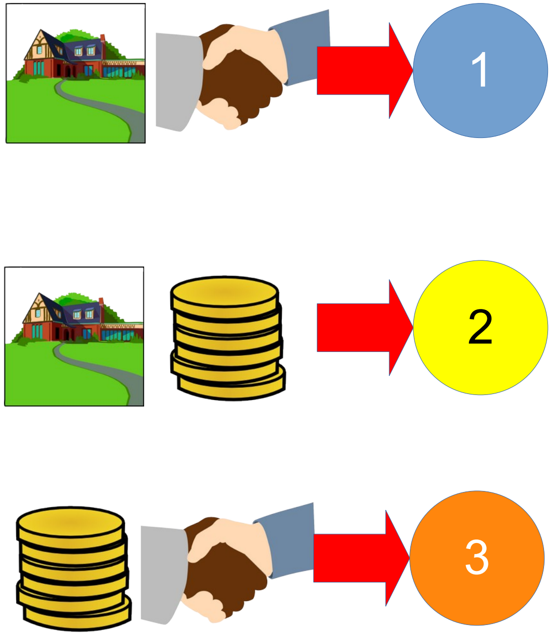




Pattern Set Mining

Conceptual/predictive clustering: rules predict clusters





Overview

Itemset mining: Background, In CP, In SAT

Pattern set mining: Background, **In CP**, In MIP

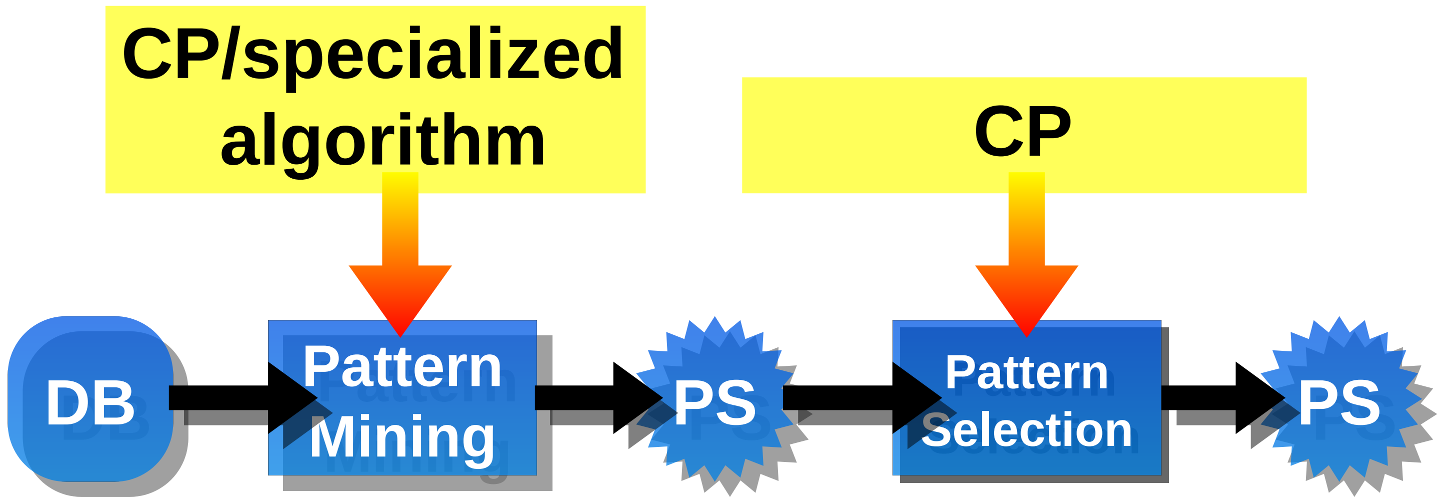
Decision trees: Background, In SAT, MIP, CP

Constraint Programming in Pattern Set Mining

Two approaches

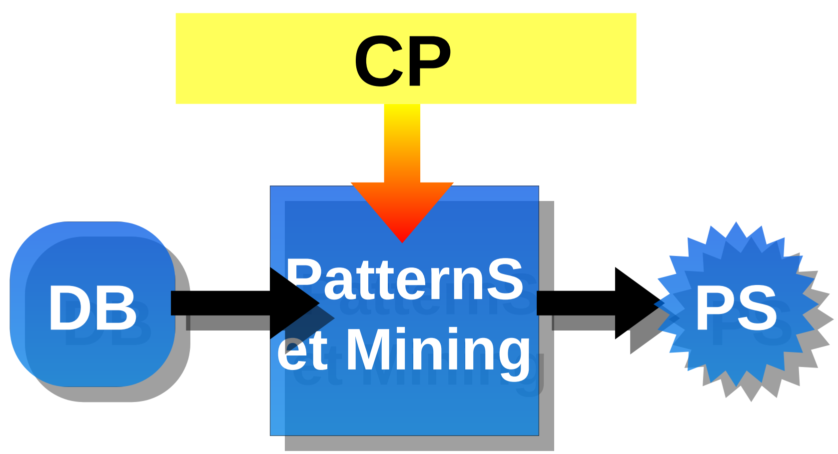
Hybrid Formulation

Step-wise



Pure CP Formulation

One step



K-Pattern Set Mining: One Step - CP

L. De Raedt, S. Nijssen, T. Guns. k-Pattern Set Mining under Constraints. IEEE TKDE, 2013.

k sets of pattern variables ($pattern_1, \ldots, pattern_k$)

- Constraints on each of the *k* patterns

  $closed(pattern_i)$

  $freq(pattern_i) > minsup$, $freq(pattern_i) < maxsup$

  $size(pattern_i) < maxsize$, $size(pattern_i) > minsize$

- Constraints on pattern sets, for instance

$$\min_{1 \le i < j \le k} distance(pattern_i, pattern_j) < maxd$$

- Optimization criteria on pattern set, for instance, accuracy

All mapped into “small” (not global) constraints in CP

K-Pattern Set Mining: One Step - CP

L. De Raedt, S. Nijssen, T. Guns. k-Pattern Set Mining under Constraints. IEEE TKDE, 2013.

Problems that can be modeled using these primitives:

- *k*-Tiling
- *k*-Concept learning
- *k*-Conceptual clustering (with no overlap between clusters)
- Redescription mining

Overview

Itemset mining: Background, In CP, In SAT

Pattern set mining: Background, In CP, **In MIP**

Decision trees: Background, In SAT, MIP, CP

K-Pattern Set Mining: Two Step - MIP

A. Ouali, A. Zimmermann, S. Loudni, Y. Lebbah , B. Cremilleux, P. Boizumault, L. Loukil. Integer Linear Programming for Pattern Set Mining; with an Application to Tiling. PAKDD 2017.

A 2-phase approach, in which LCM is used to solve the first phase and MIP is used to solve the second phase

In principle could support a number of pattern set mining settings; only evaluated on *tiling*

$\mathcal{D}$	Items	Trans.	k	Recall					Time (s)				
				(1)	(2)	(3)	(4)	(5)	(1)	(2)	(3)	(4)	(5)
Zoo-1	34	101	3	0.48	0.38	0.38	0.48	0.46	8.99	5.87	1.57	2,418	
			4	0.57	0.47	0.48	TO	0.56	14.38	4.53	3.67	TO	
			5	0.63	0.54	0.52	TO	0.62	25.11	4.49	2.99	TO	
			6	0.68	0.55	0.55	TO	0.67	34.96	10.82	2.98	TO	
			7	0.73	0.57	0.58	TO	0.71	22.85	5.81	3.49	TO	
			8	0.78	0.68	0.61	TO	0.75	25.56	6.96	3.16	TO	
			9	0.82	0.69	0.64	TO	0.79	35.07	8.04	3.59	TO	
			10	0.85	0.69	0.66	TO	0.82	26.21	6.91	3.23	TO	0.99
			3	0.43	0.42	0.35	TO	0.42	299.91	50.92	152.54	TO	

(1) 2-Phase standard tiling
(2), (3) variations
(4) Gecode 1-Phase
(5) Specialized algorithm

K-Pattern Set Mining: One Step – MIP / Greedy

J. Aoga, S. Nijssen, P. Schaus. Modeling Pattern Set Mining using Boolean Circuits. CP 2019.

Inspired by *neural networks,* a modeling language is used based on *parameter learning in boolean circuits*

Allows for modeling:

- Tiling
- Conceptual clustering (with overlap)
- Rule learning (ordered/unordered)

Solvers currently supported:

- MIP solvers
- Greedy solvers
- More to come

K-Pattern Set Mining: One Step – MIP / Greedy

J. Aoga, S. Nijssen, P. Schaus. Modeling Pattern Set Mining using Boolean Circuits. CP 2019.

	i_1	i_2	i_3	i_4	i_5	
1	1	1	1	0	1	+
2	1	1	0	1	1	−
3	1	0	1	1	1	+
4	1	0	1	1	0	+

Database

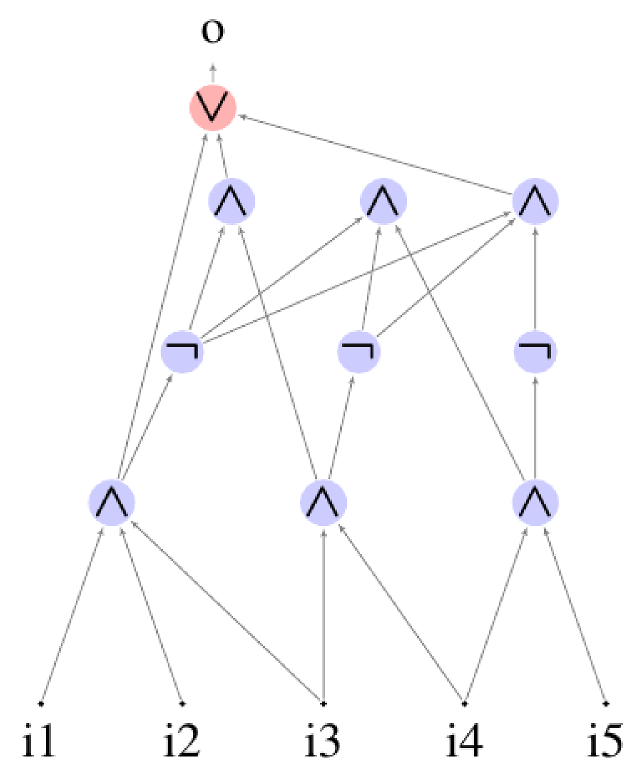
IF $\{i_1, i_2, i_3\}$ **THEN** predict +

ELSE IF $\{i_3, i_4\}$ **THEN** predict +

ELSE IF $\{i_4, i_5\}$ **THEN** predict −

ELSE (default case) **THEN** predict +

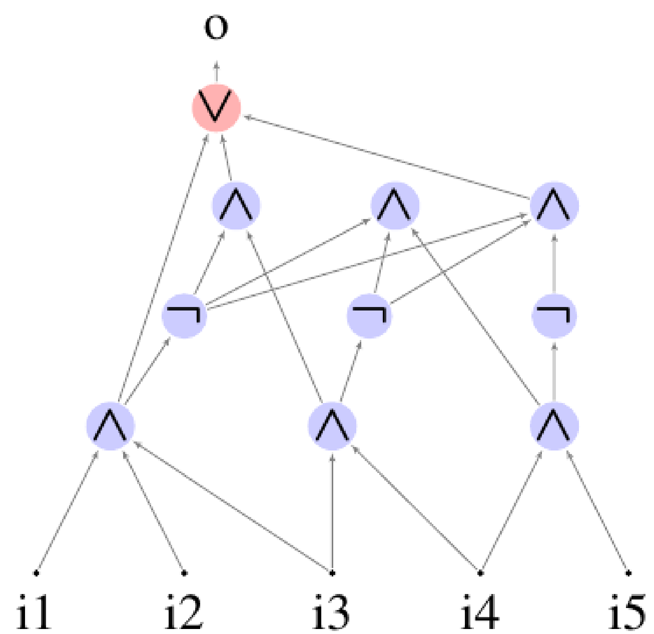
Rule list



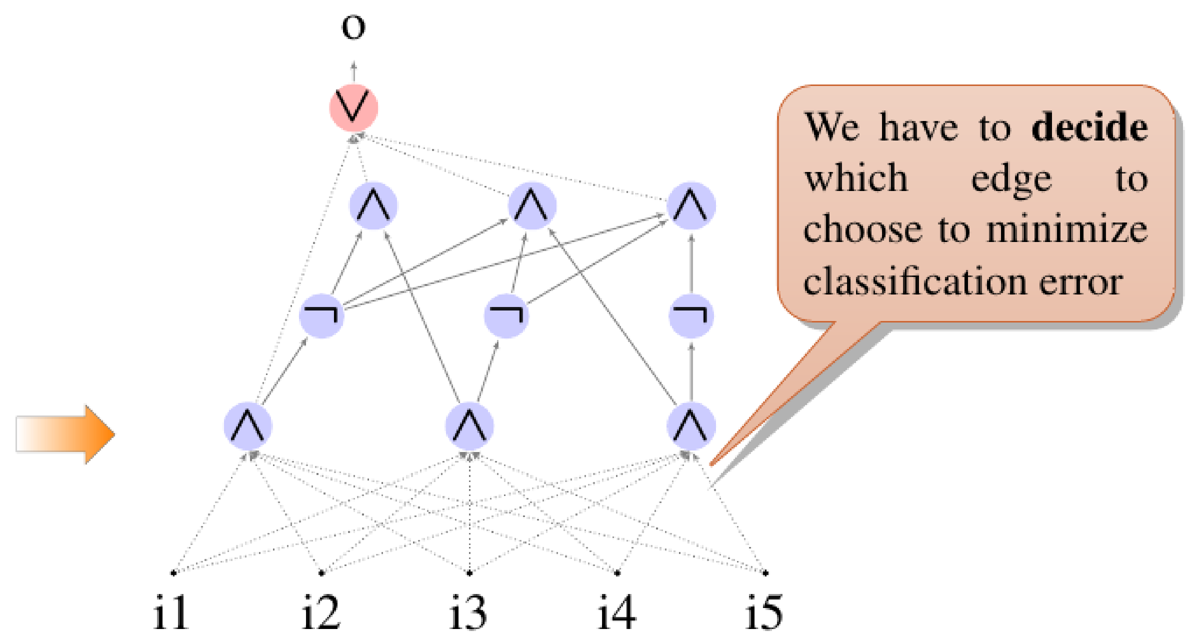
Representation of Boolean circuit

K-Pattern Set Mining: One Step – MIP / Greedy

J. Aoga, S. Nijssen, P. Schaus. Modeling Pattern Set Mining using Boolean Circuits. CP 2019.



Equivalent model Boolean Circuit



Boolean Circuit Model

K-Pattern Set Mining: One Step – MIP / Greedy

J. Aoga, S. Nijssen, P. Schaus. Modeling Pattern Set Mining using Boolean Circuits. CP 2019.

methods	Audi.	Aust.	HeCl.	Hepa.	KrKp.	Lymp.	Mush.	PrTu.	Soyb.	Spli.	TTT.	Vote	Zoo
	b) Accuracies over training sets												
MIP4CL-α	**1.0**	**0.92**	0.89	0.98	0.87	0.97	0.55	0.89	0.97	-	**0.90**	0.98	**1.0**
MIP4CL+α	**1.0**	0.91	**1.0**	**1.0**	0.93	**1.0**	**1.0**	**0.91**	**1.0**	**0.85**	0.77	**1.0**	**1.0**
G4CL	0.73	0.45	0.46	0.19	0.47	0.46	0.49	0.76	0.86	0.48	0.35	0.37	0.59
KPATT	**1.0**	-	-	-	-	-	-	0.89	0.97	-	0.82	-	**1.0**
	d) Running time (in second) - TO≡ Timeout												
MIP4CL-α	26.09	TO	TO	TO	TO	TO	TO	TO	TO	-	TO	TO	1.65
MIP4CL+α	**5.81**	TO	**2682.90**	**1.99**	TO	**2.50**	**251**	TO	**915.09**	TO	TO	**17.32**	**0.66**
KPATT	20	-	-	-	-	-	-	TO	1730.30	-	TO	-	3.29

Overview

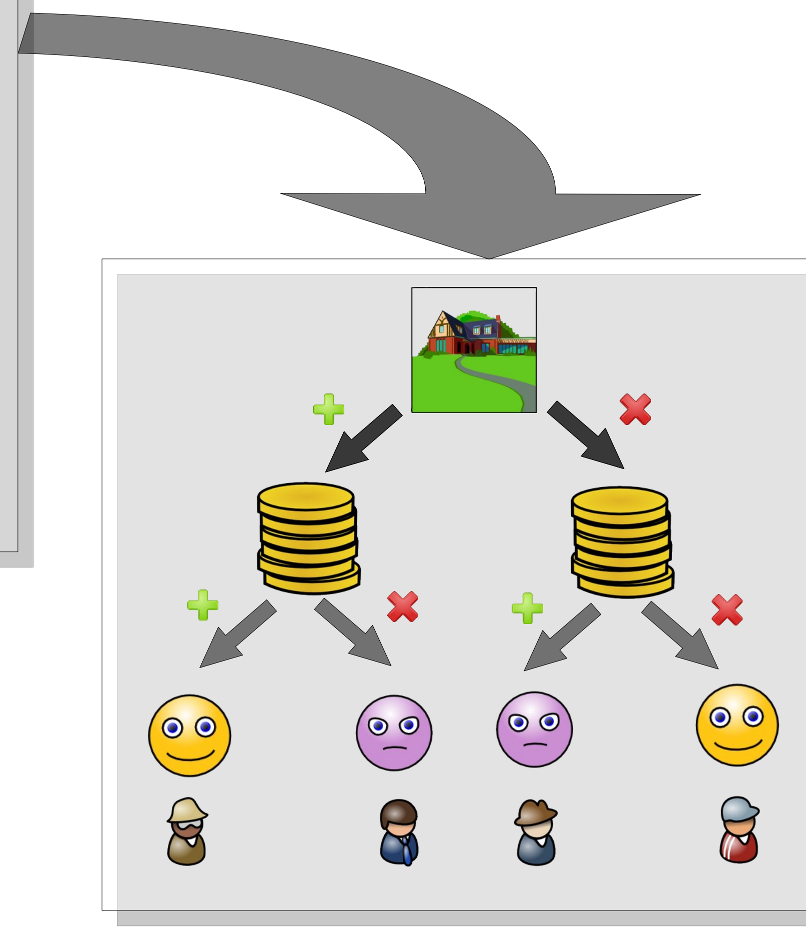
Itemset mining: Background, In CP, In SAT

Pattern set mining: Background, In CP, In MIP

Decision trees: **Background**, In SAT, MIP, CP

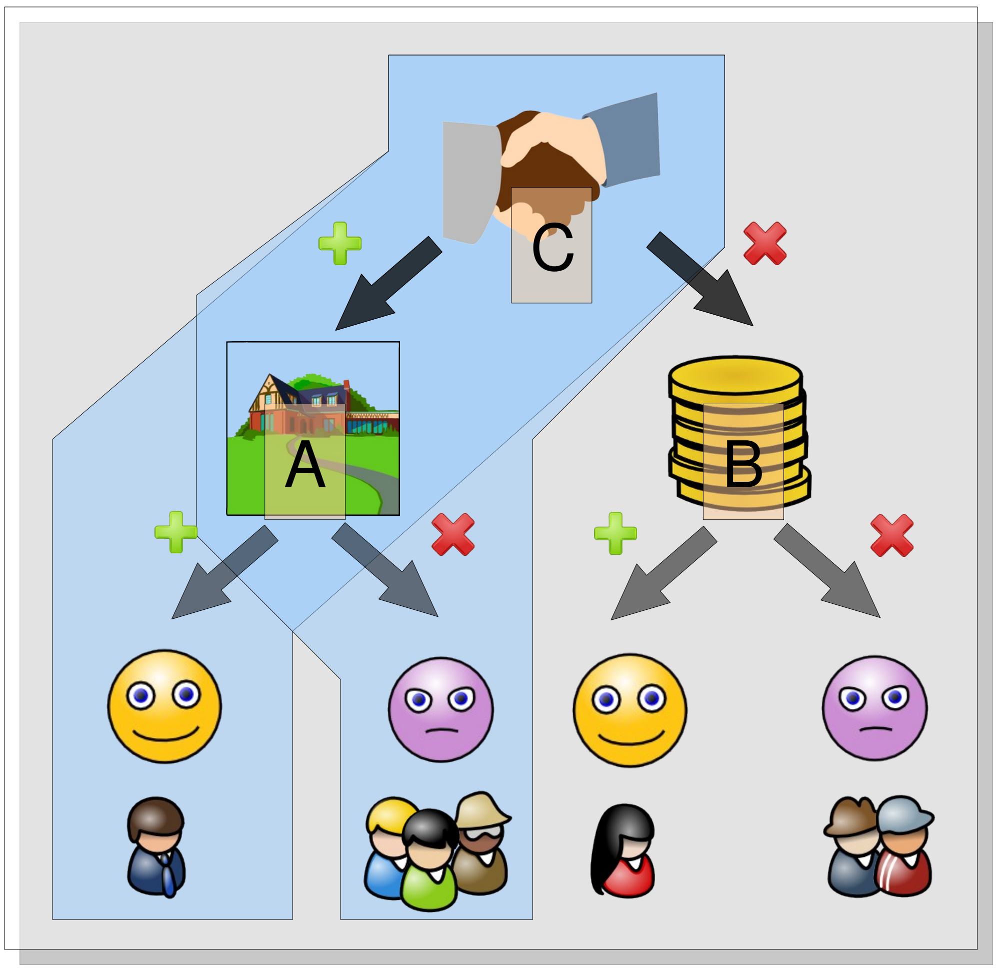
Decision Trees

Dataset



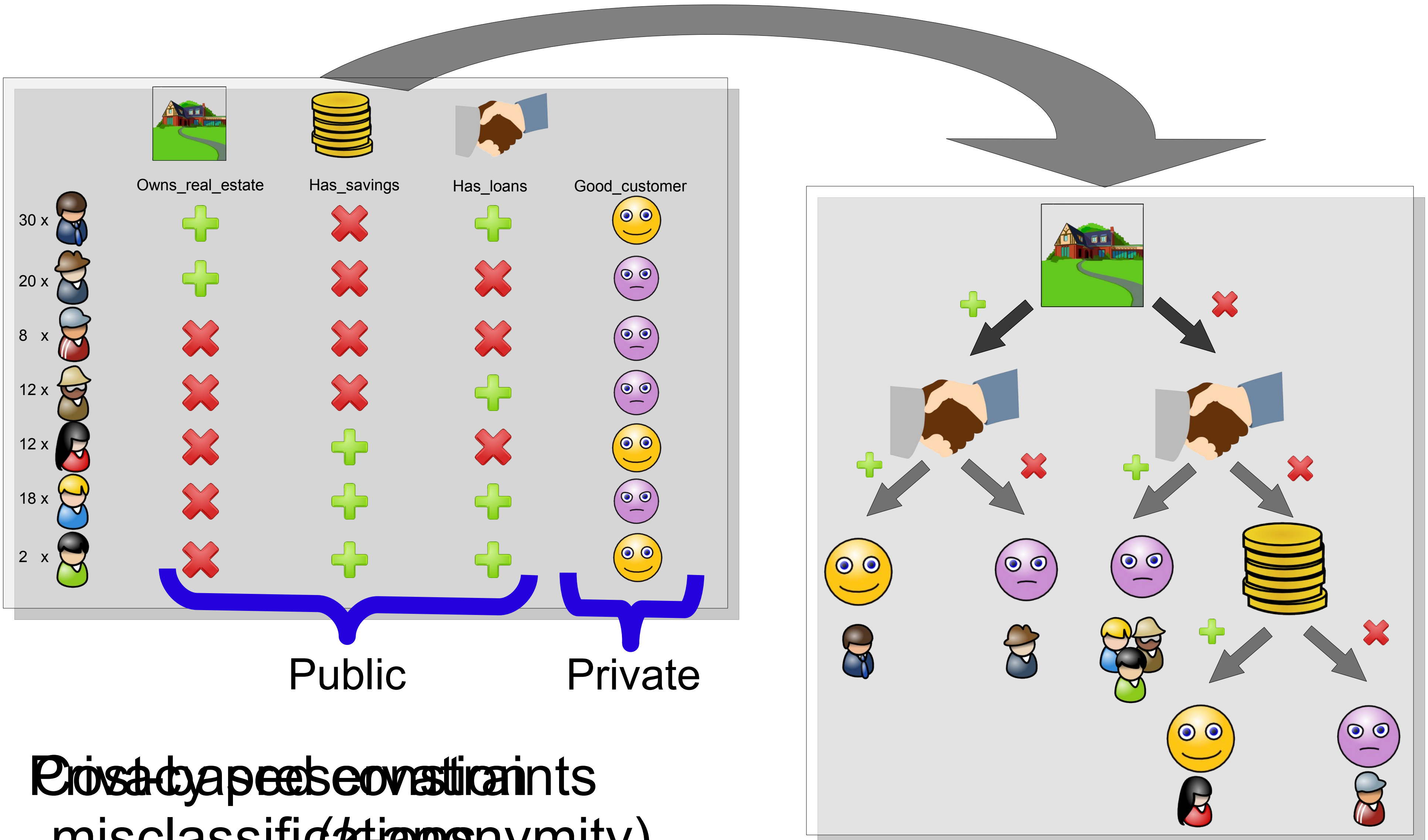
Decision Tree

Decision Tree = Pattern Set



Itemset	Tids
{ C, A }	
{ C, ¬A }	
{ ¬C, B }	
{ ¬C, ¬B }	

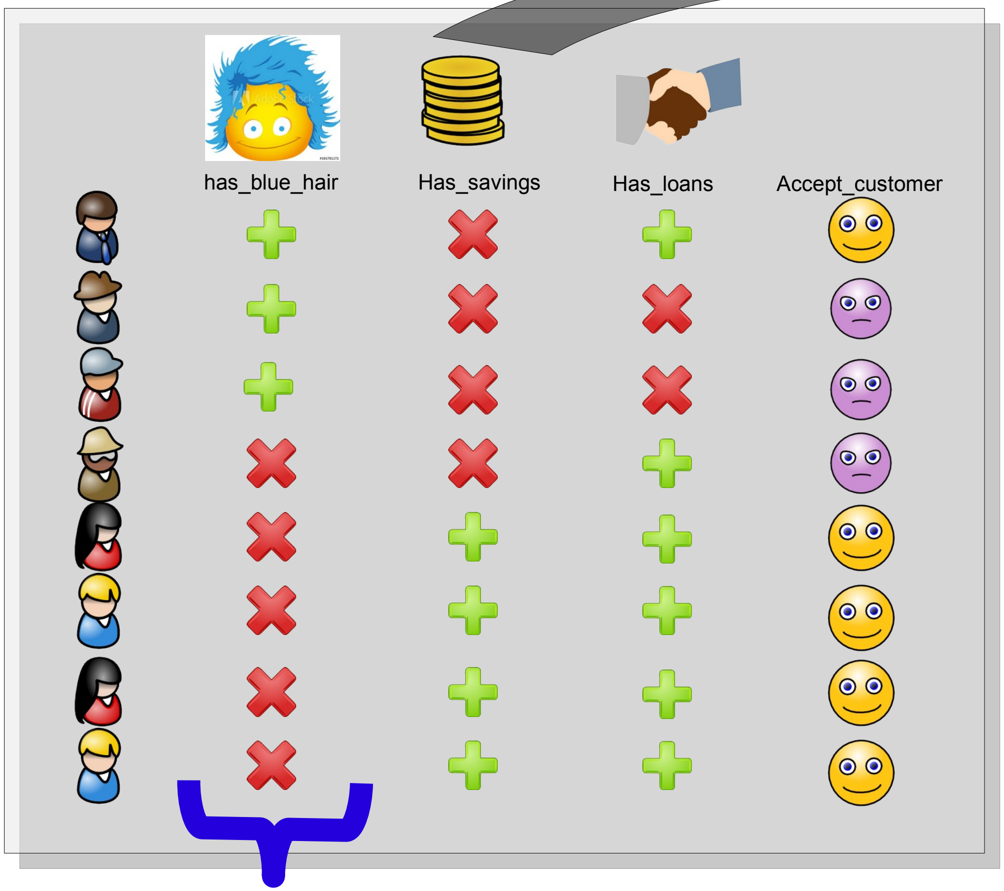
Prediction: Why Constraints?



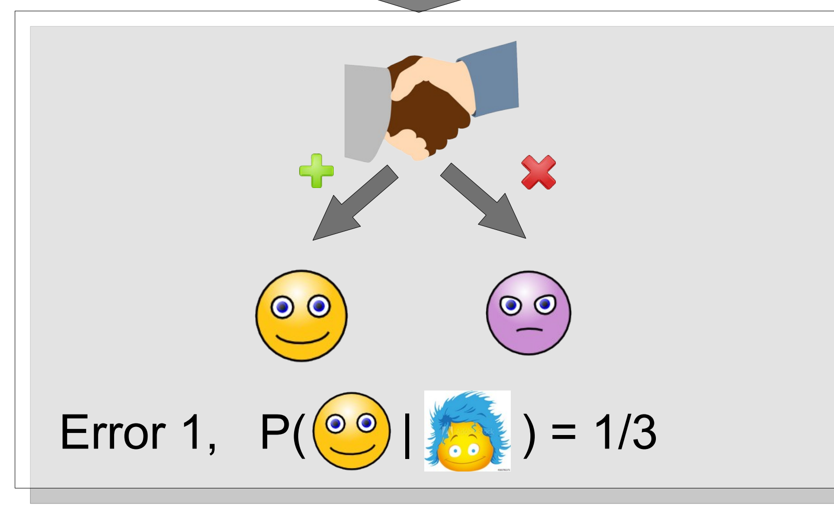
Cost-based: misclassifications

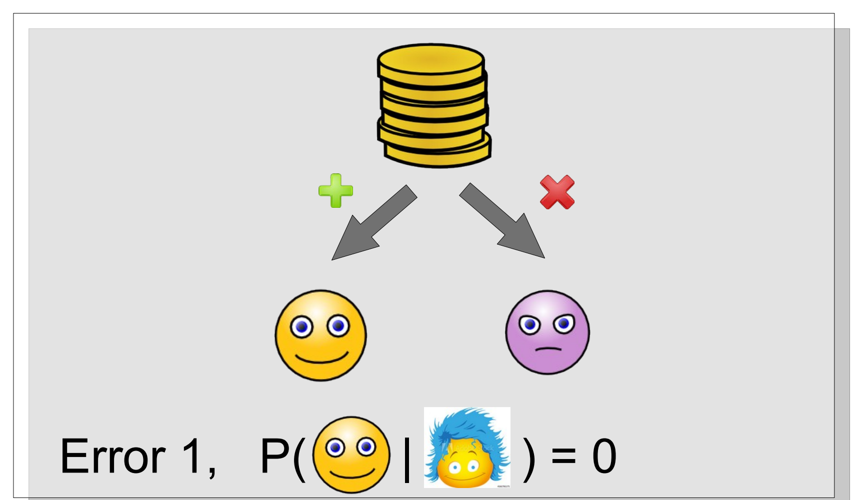
Privacy-based constraints (k-anonymity)

Prediction: Why Constraints?



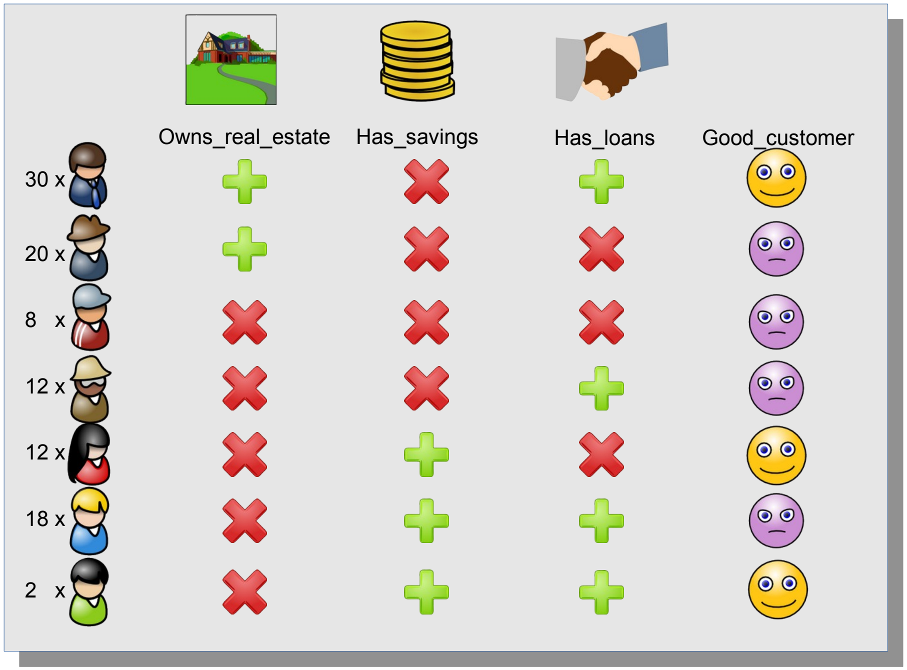
Protected

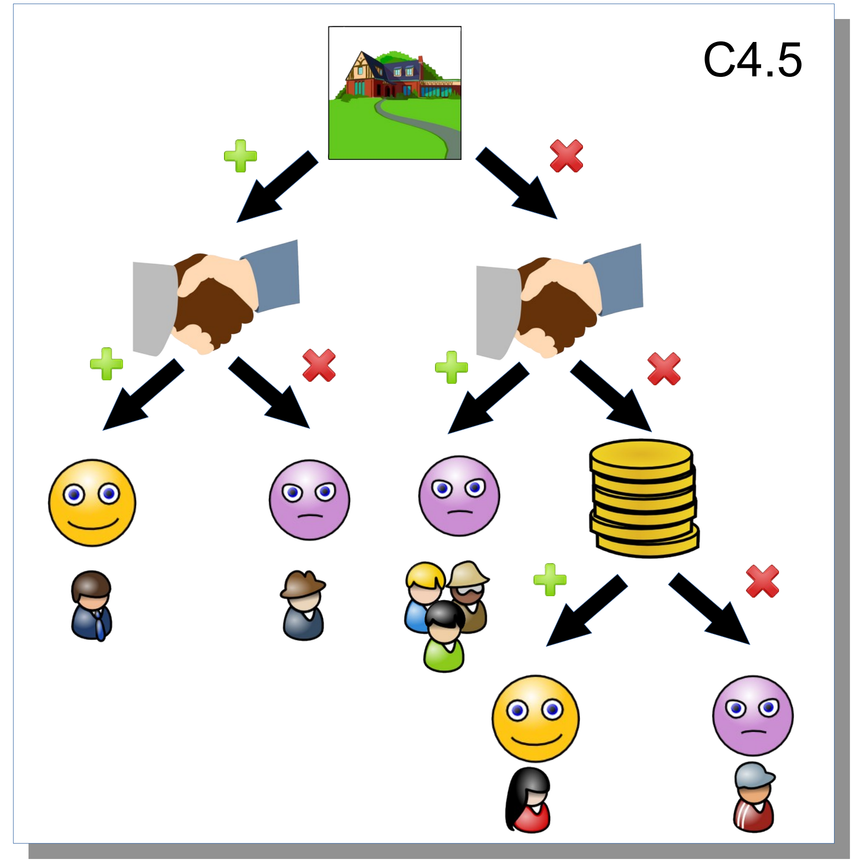


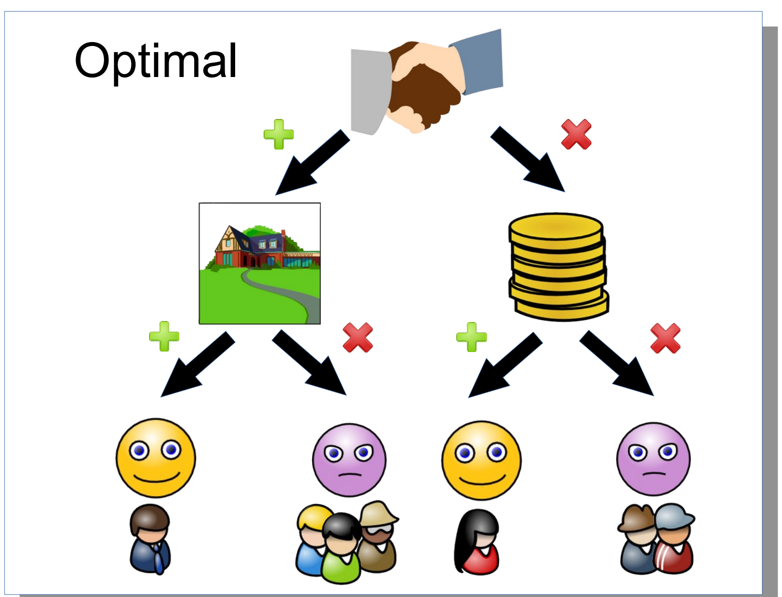


Fairness of predictions
towards protected groups

Prediction: Why Optimal Solving?



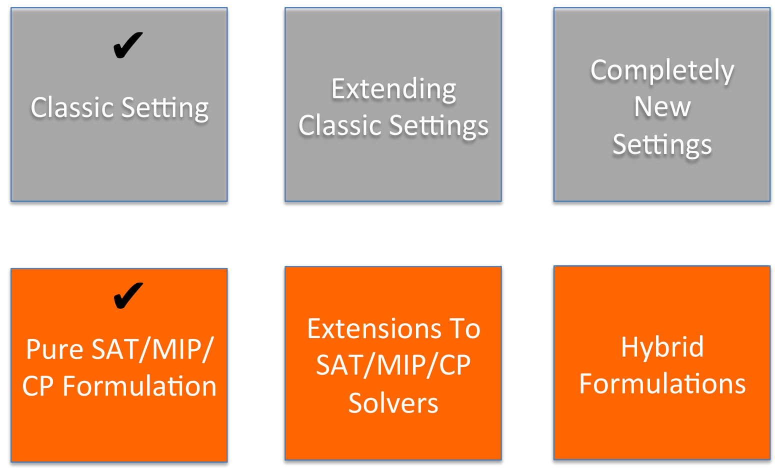




C4.5 cannot compute trees that optimize a desirable criterion, such as simplicity of the model

Computing optimal trees under constraints is NP complete

Overview



Starting with classic settings

Overview

Itemset mining: Background, In CP, In SAT

Pattern set mining: Background, In CP, In MIP

Decision trees: Background, **In SAT**, MIP, CP

SAT-based Approaches

C. Bessiere, E. Hebrard, B. O'Sullivan. Minimising Decision Tree Size as Combinatorial Optimisation. CP, 2009.

N. Narodytska, A. Ignatiev, F. Pereira, J. Marques-Silva. Learning Optimal Decision Trees with SAT. IJCAI, 2018.

Solve the following problem:

Given a parameter *K*, a dataset *D*

Find a tree of size *K*, such that the error of this tree on *D* is <u>**zero**</u>

… as SAT solvers solve satisfaction and not optimization problems

Overview

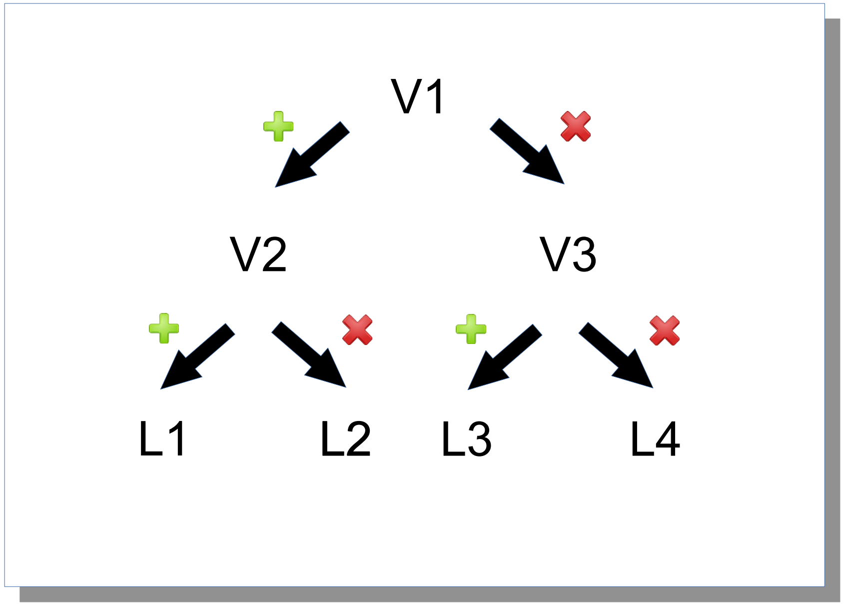
Itemset mining: Background, In CP, In SAT

Pattern set mining: Background, In CP, In MIP

Decision trees: Background, In SAT, **MIP**, CP

MIP-based Approaches

- Impose depth constraint
- Use variables for
  - Decisions in the internal nodes of the tree
  - Labels in the leafs of the tree
  - Examples in the training data covered by each leaf



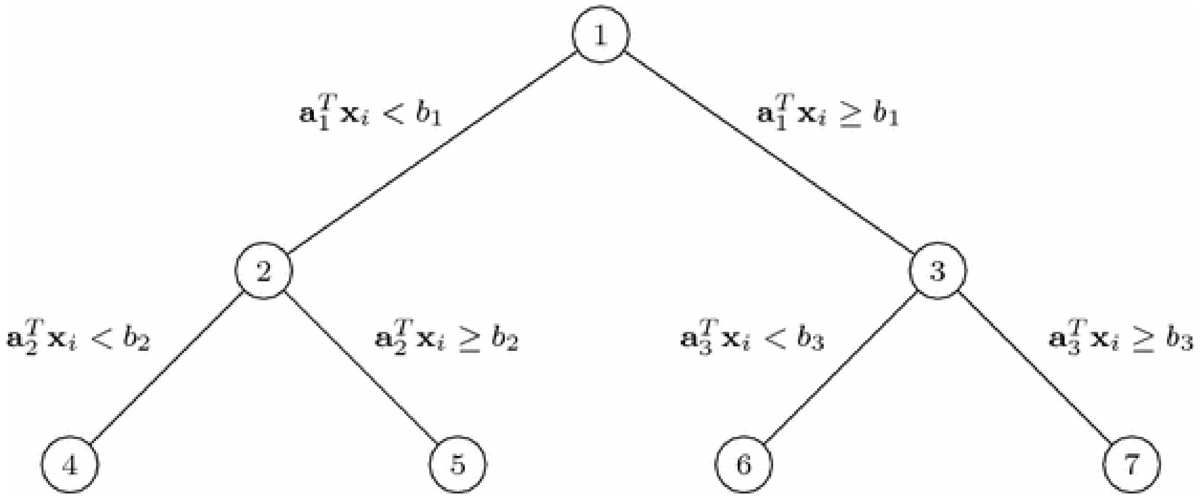
- Encode cover of paths, loss function using linear constraints

OCT

D. Bertsimas, J. Dunn. Optimal classification trees. Mlj, 2017.

Finds the most accurate classification tree

A vector **a** of variables for each node to choose the feature used in that node, summing up to 1



"Big-M" approach used to determine the leaf for each training example: constraints for every training example

BinOCT

S. Verwer, Y. Zhang. Learning optimal classification trees using a binary linear program formulation. AAAI, 2019.

Finds the most accurate classification tree

Formulation in which we don't have a set of constraints for every training example

→ Model with a smaller number of constraints

→ Can find better solutions within a given amount of run time

Overview

Classic Setting	Extending Classic Settings	✔ Completely New Settings
✔ Pure SAT/MIP/CP Formulation	Extensions To SAT/MIP/CP Solvers	Hybrid Formulations

New Setting: Fairness

MIP for Fair Decision Trees

S. Aghaei, M.J. Azizi, P. Vayanos. Learning Optimal and Fair Decision Trees for Non-Discriminative Decision-Making. AAAI, 2019.

Finds trees that optimize a sum of:

- Accuracy
- Disparate impact
  → knowing a sensitive attribute should not change the distribution of predictions
- Disparate treatment
  → similar examples over all features should receive a similar classification

Basic formulation similar to OCT + additional variables / constraints for the additional terms in the loss function

Overview

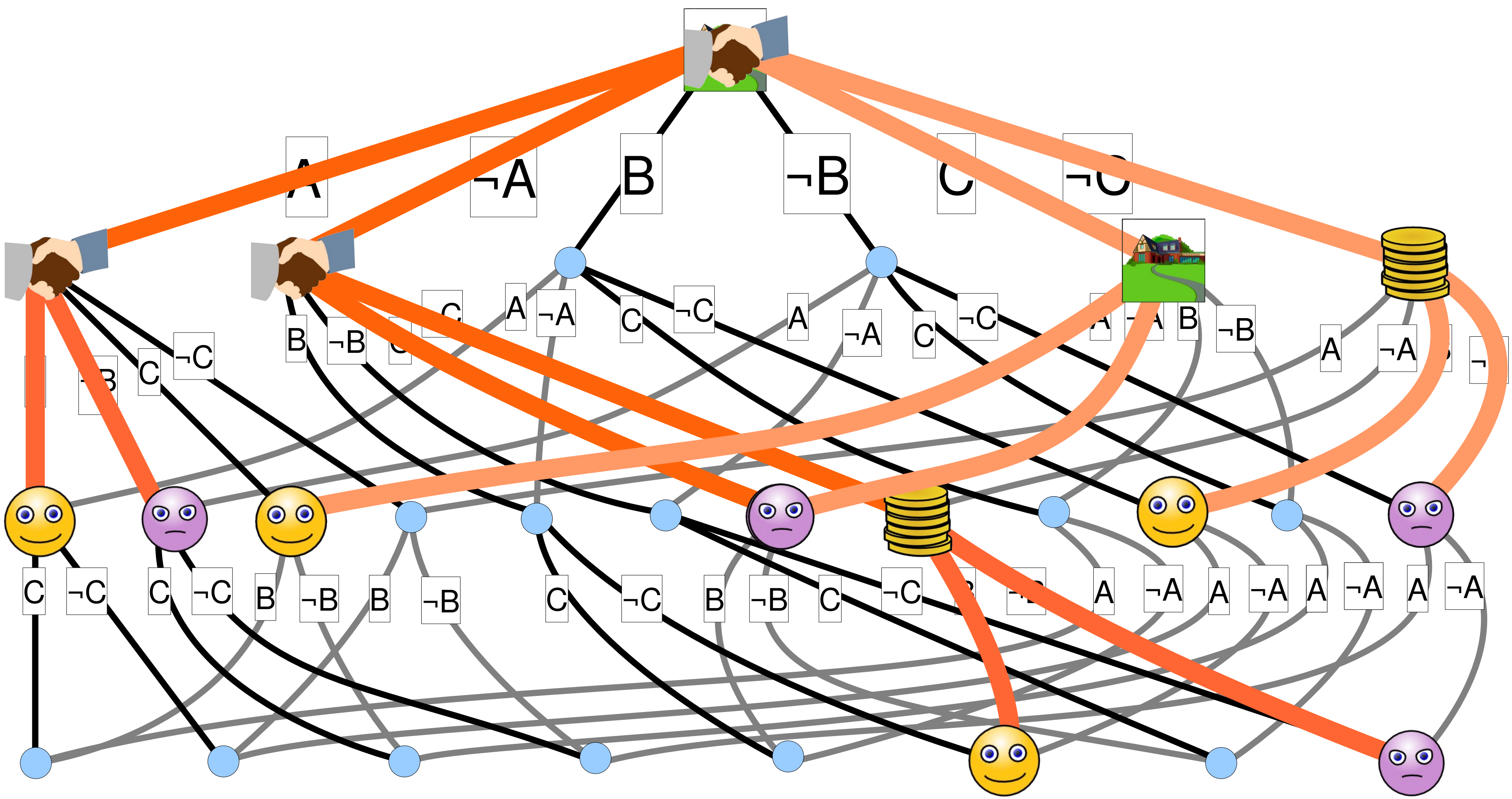
Itemset mining: Background, In CP, In SAT

Pattern set mining: Background, In CP, In MIP

Decision trees: Background, In SAT, MIP, **CP**

DL8: Lattice of Itemsets

S. Nijssen, E. Fromont. Mining optimal decision trees from itemset lattices. KDD, 2017.



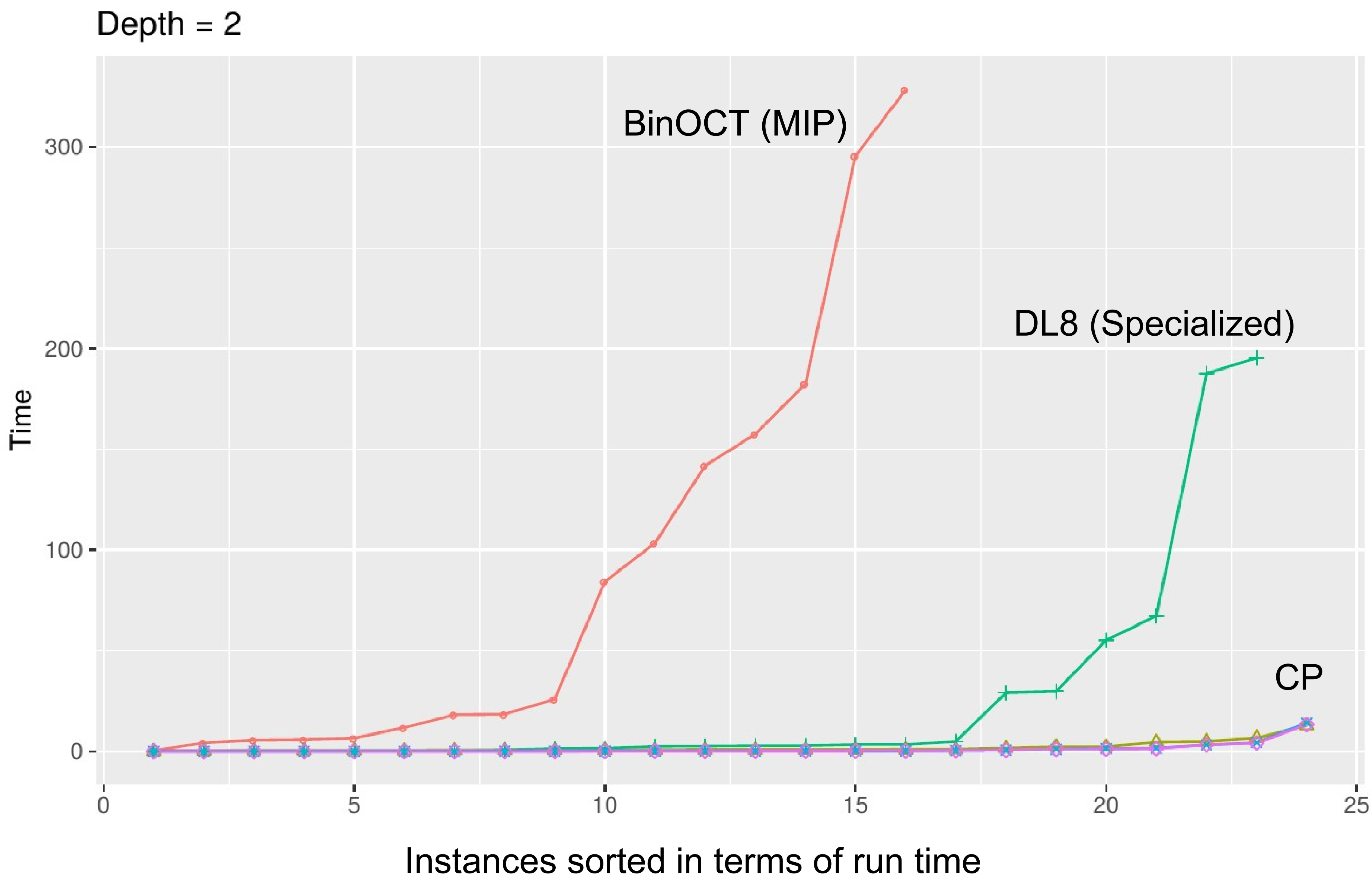
Decision Trees are hidden in the lattice of itemsets

CP-based Approach

H. Verhaeghe, S. Nijssen, G. Pesant, C.G. Quimper, P. Schaus. Learning Optimal Decision Trees using Constraint Programming. CP, 2019.

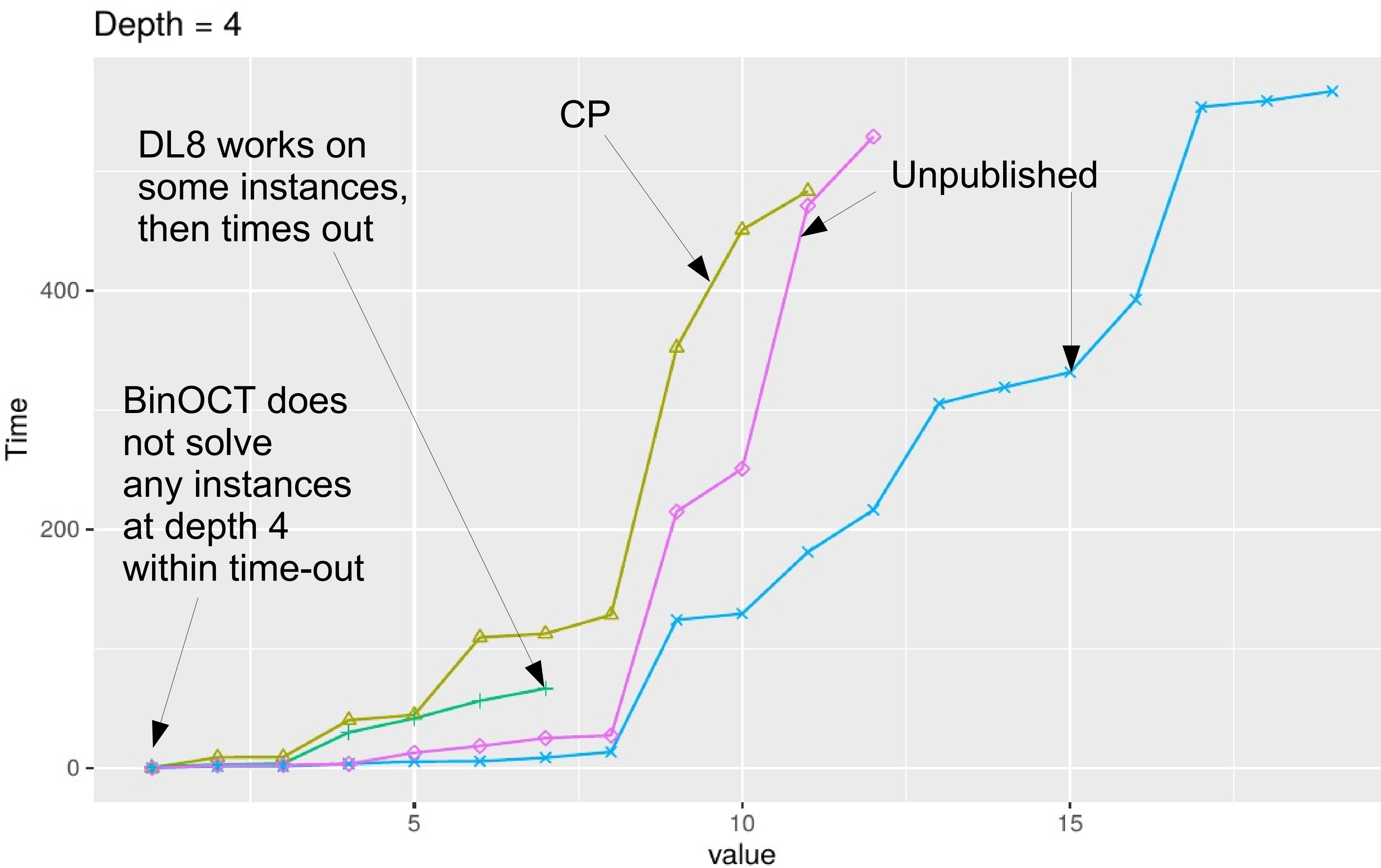
- Builds on ideas present in DL8
- Imposes depth constraint
- Similar variables as in MIP
- However,
    - Use CoverSize constraint for itemset mining to calculate cover
    - Use AND/OR search
    - Cache partial results for itemsets

Results



Depth = 2
BinOCT (MIP)
DL8 (Specialized)
CP
Time
300
200
100
0
0
5
10
15
20
25
Instances sorted in terms of run time

Results



Instances sorted in terms of run time

Pattern (Set) Conclusions

Pattern mining, rule learning and decision tree learning problems have successfully been modeled using MIP, CP and SAT

CP solvers often obtain better computational performance, provided specialized global constraints are used

Significant potential for the addition of constraints related to fairness, explainability, privacy

References Pattern (Set) Mining

L. De Raedt, T. Guns, S. Nijssen. Constraint programming for itemset mining. KDD 2008.

S. Nijssen, T. Guns. Integrating constraint programming and itemset mining. ECML PKDD 2010.

M. Maamar, N. Lazaar, S. Loudni, Y. Lebbah. A global constraint for closed itemset mining. CP 2016.

P. Schaus, J. Aoga, T. Guns. CoverSize: A global constraint for frequence-based itemset mining. CP 2017.

S. Jabbour, L. Sais, Y. Salhi. The top-k frequent closed itemset mining using top-k sat problem. ECML PKDD 2013.

R. Henriques, I. Lynce, V. Manquinho. On When and How to use SAT to Mine Frequent Itemsets. ArXiv:1207.6253. 2012.

T. Toda, T. Soh. Implementing Efficient All Solutions SAT Solvers. ACM Journal of Experimental Algorithmics, 2016.

M. Järvisolo. Itemset Set Mining as a Challenge for Anser Set Enumeration. LPNMR 2011.

S. Nijssen, T. Guns, L. De Raedt. Correlated itemset mining in ROC space. KDD 2009.

B. Negrevergne, A. Dries, T. Guns, S. Nijssen. Dominance programming for itemset mining. ICDM 2013.

L. De Raedt, S. Nijssen, T. Guns. k-Pattern Set Mining under Constraints. IEEE TKDE, 2013.

A. Ouali, A. Zimmermann, S. Loudni, Y. Lebbah1 , B. Cremilleux, P. Boizumault, L. Loukil. Integer Linear Programming for Pattern Set Mining; with an Application to Tiling. PAKDD 2017.

OUTLINE

Previously

- itemsets
- pattern sets and decision trees

Up next: structured pattern mining!

- Sequence mining
- Multi-relational patterns
- *Interestingness* as objective function
- Wrap up of pattern mining

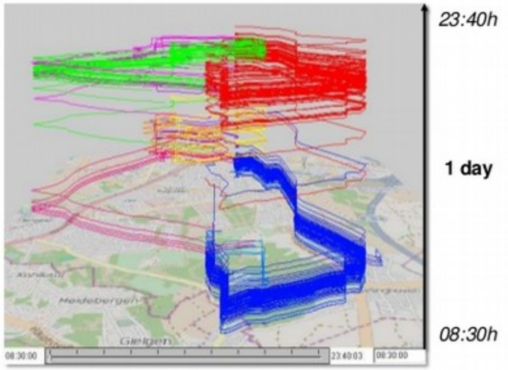
Sequential data

Example:

<Home, Work, Restaurant, Work, Home>

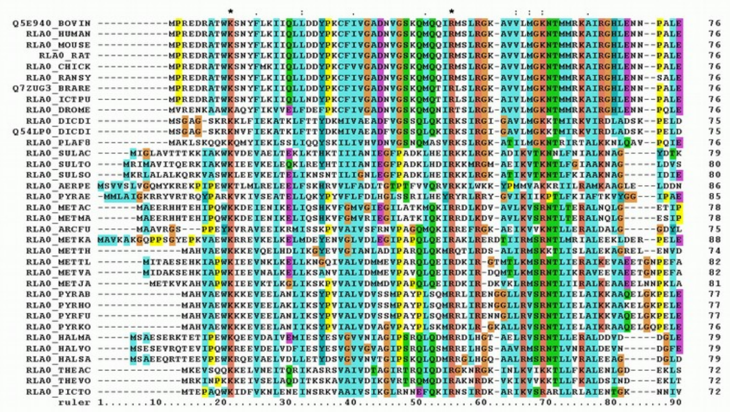
<Home, Work, Shops, Restaurant, Home>

...



Many applications:

- User mobility mining
- Web usage mining
- Event monitoring
- Biological sequence mining (DNA, Amino acids)
- ...



Sequence mining

Cover

= subsequence relation

= ordered matching

Pattern: <H, G, ?, ?, ?>

T1: <S,B,H,R,G,H,M>

T2: <S,G,H,W,L,W,M>

T3: <R,H,W,H,D,G,H>

multiple embeddings possible:

T3: <R,**H**,W,**H**,D,G,H>

Constraints

<u>Many constraints</u>, we identify four categories:

- Constraints on syntax:

  size, regular expression, ...

- Constraints on data covered:

  min_support, max_support, discriminative, ...

- Constraints on the relationship to other patterns

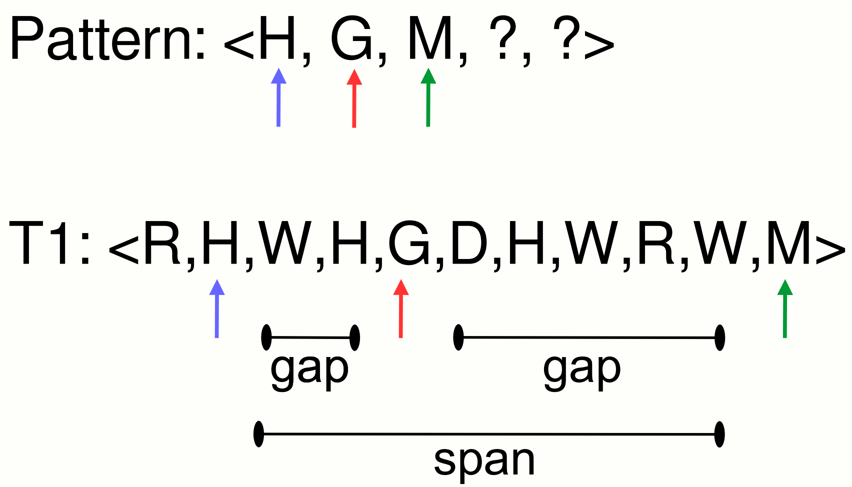
  closed, maximal, relevant, multi-objective, ...

- **new** Constraints on cover relation:

  max_gap, min_gap, max_span

Hard-coded in specialised algorithms...

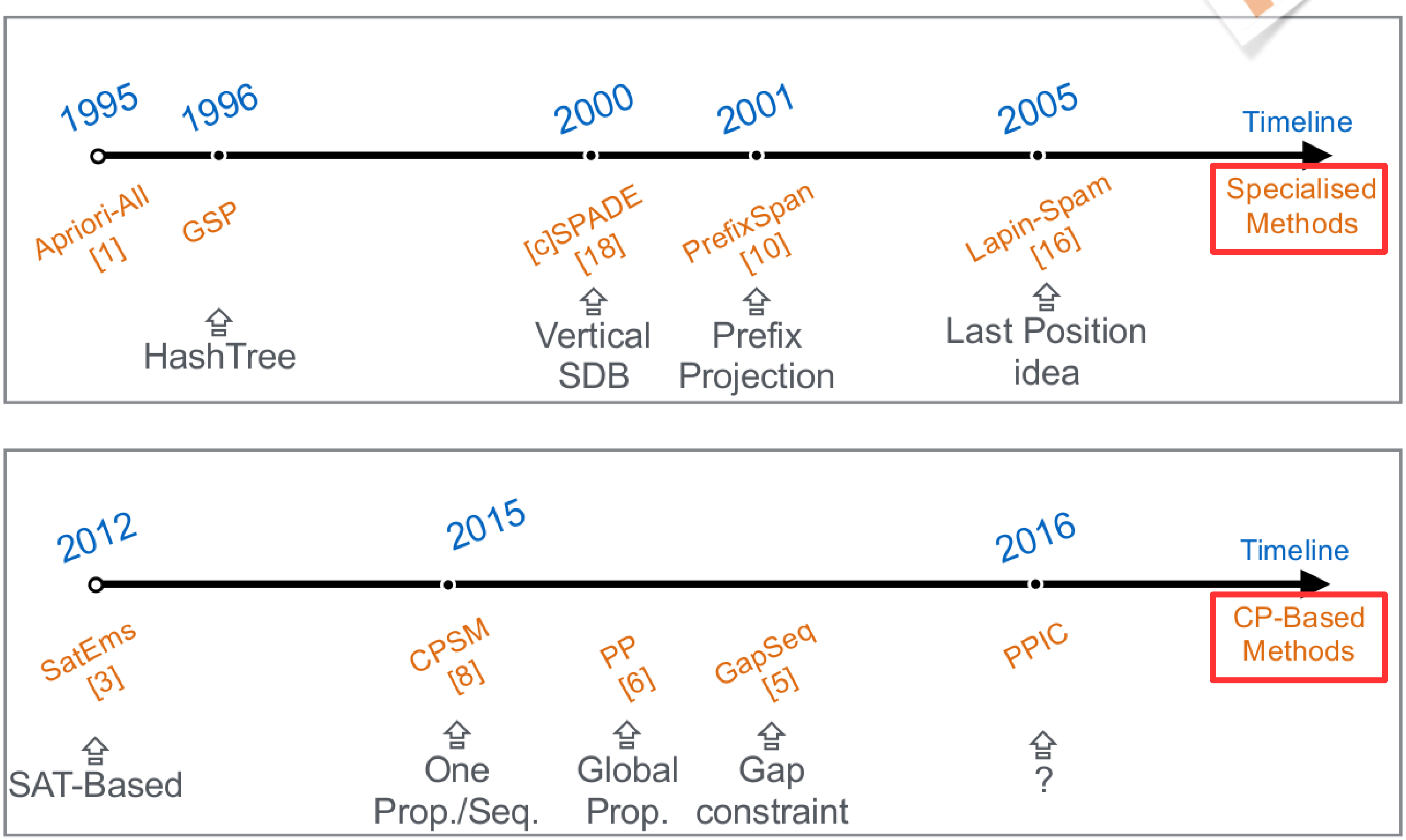
Constraints on cover relation



- min/max gap constraint (ex. max gap = 20 base pairs)
- min/max span constraint (ex. max span = 2 hours)

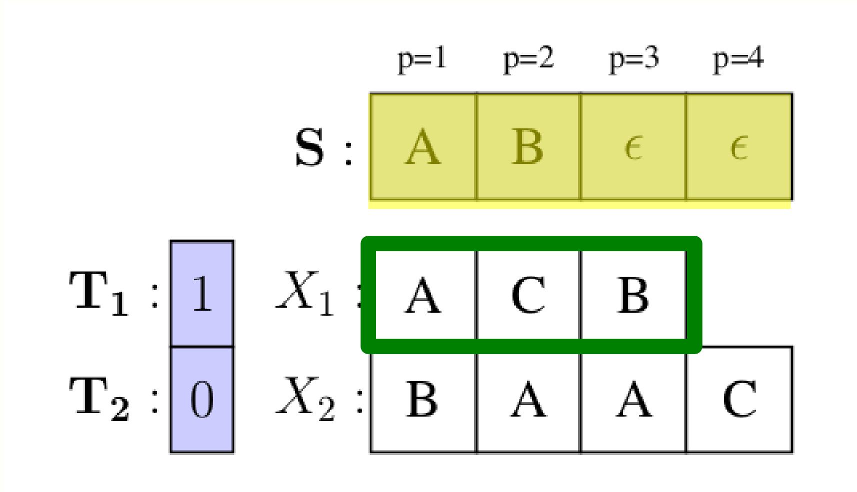
When distance indicative of relatedness

Related Works



[source: John Aoga]

Sequence mining in CP



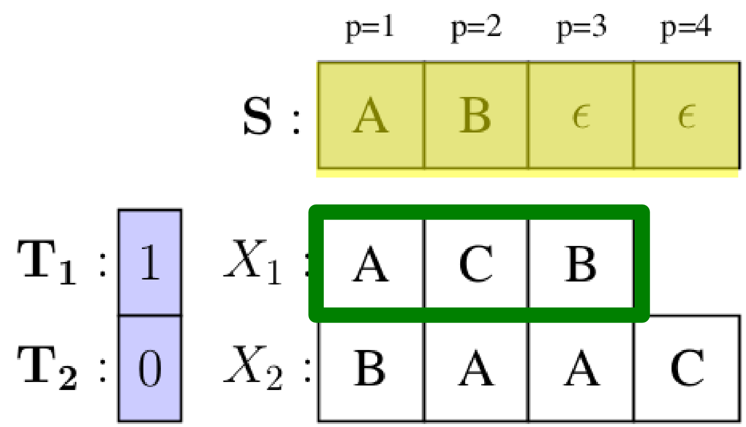
Cover:

$$\forall T_t : T_t = 1 \Leftrightarrow exist-embedding\left(S, \boxed{X_t}\right)$$

Frequency:

$$\sum T_t \geq Freq$$

CP1: pure modeling



Cover:

$$T_t = 1 \Leftrightarrow \exists (e_1 .. e_n) : e_1 < \ldots < e_n \wedge \forall_j S[j] = \boxed{X_t}[e_j]$$

Frequency:

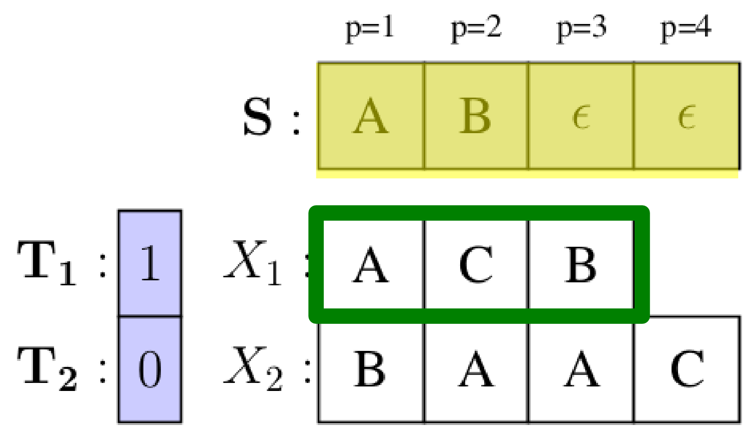
$$\sum T_t \geq Freq$$

needs O(nm) integer variables *e* (for each transaction and position) and 2nd order logic (backjumping search, <u>not pure</u>...)

[B. Negrevergne, T. Guns, CPAIOR 15]

CP2: fine-grained global constraints

Extensions To SAT/MIP/CP Solvers



Cover:

$$\forall T_t : T_t = 1 \Leftrightarrow exist - embedding(S, \boxed{X_t})$$

Frequency:

$$\sum T_t \geq Freq$$

needs O(n) *exists-embedding* global constraints

[B. Negrevergne, T. Guns, CPAIOR 15]

What specialized algorithms do

P: <**H**, **G**, ?, ?, ?>

T1: <R,**H**,W,H,D,**G**,H>

T2: <S,B,**H**,R,**G**,H,M>

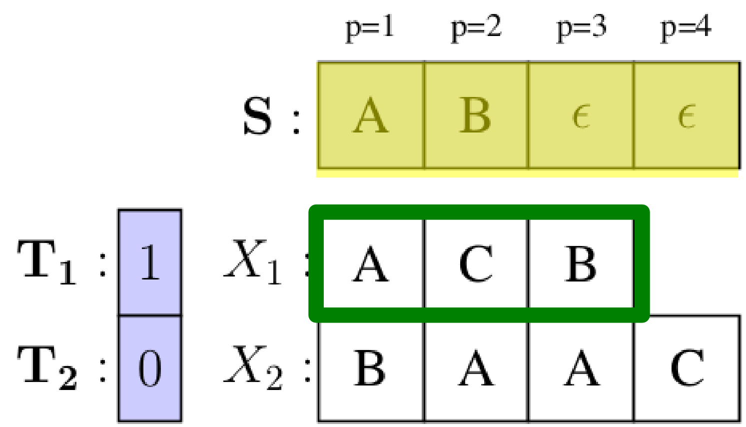
PrefixSpan:

- Linear scan of each transaction, keep only pointer to *first* match of *last* symbol (above: ↑)
- When symbol added to P, continue from pointer (incremental)
- O(1) space, single scan algorithm

[J. Han, J. Pei, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. C. Hsu. ICDE 2001]

Extensions To SAT/MIP/CP Solvers

CP3: coarse-grained global



Cover:

Frequency:

$$frequent - sequence\left(S, \boxed{X}, Freq\right)$$

one joint global constraint (note: no transaction variables)

[A. Kemmar, S. Loudni, Y. Lebbah, P. Boizumault, and T. Charnois, CP 2015]

Constraint compatibility

Constraints on:	CP1 (pure)	CP2 (fine-grained)	CP3 (coarse-grained)
- syntax	yes	yes	yes
- data	yes	yes	no
- relations to other patterns	yes	yes	no
- cover relation	yes	no	no

more efficient

(by a lot, yet not on par with specialised)

Improving CP3 (coarse-grained)

One global constraint for **all** sequences:

- while CP is a declarative system,
  the *propagator* of a constraint is a stateful imperative algorithm

$\rightarrow$ within the global constraint, you can use any advanced technique (caching, subproblem solving, custom data structures, ...)

As with global constraints for itemsets:

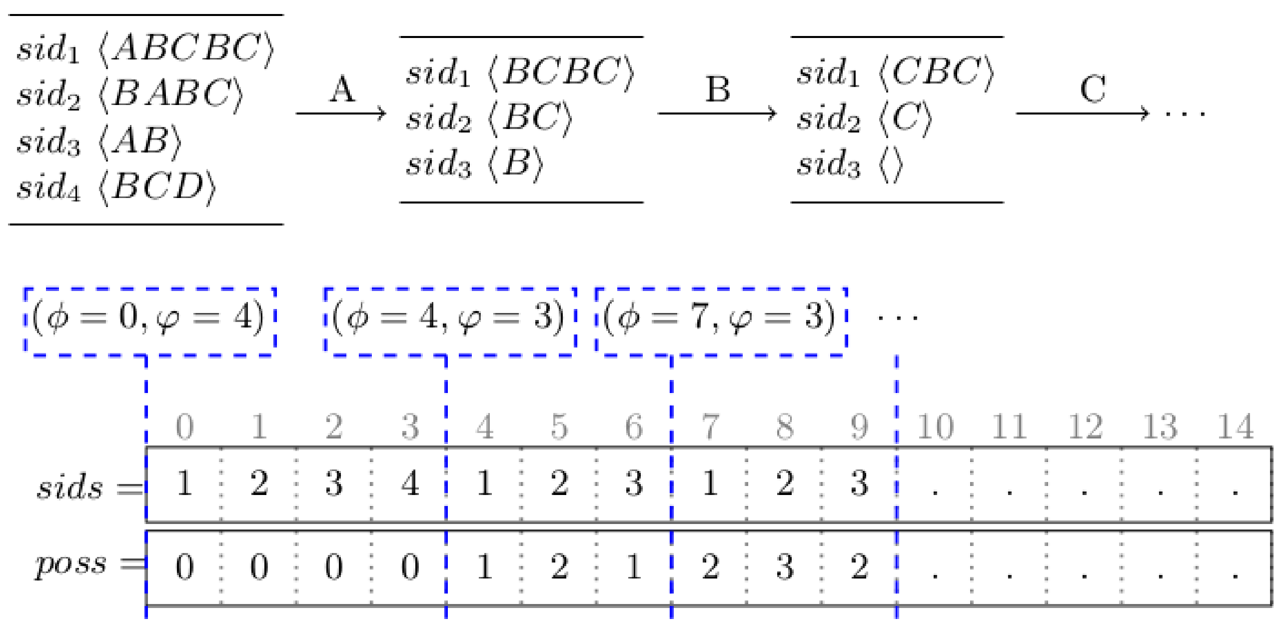
key benefit of using one global constraint =
    no need to expose the (up to millions) of transaction variables
    to the solver!

Improving CP3 (coarse-grained)

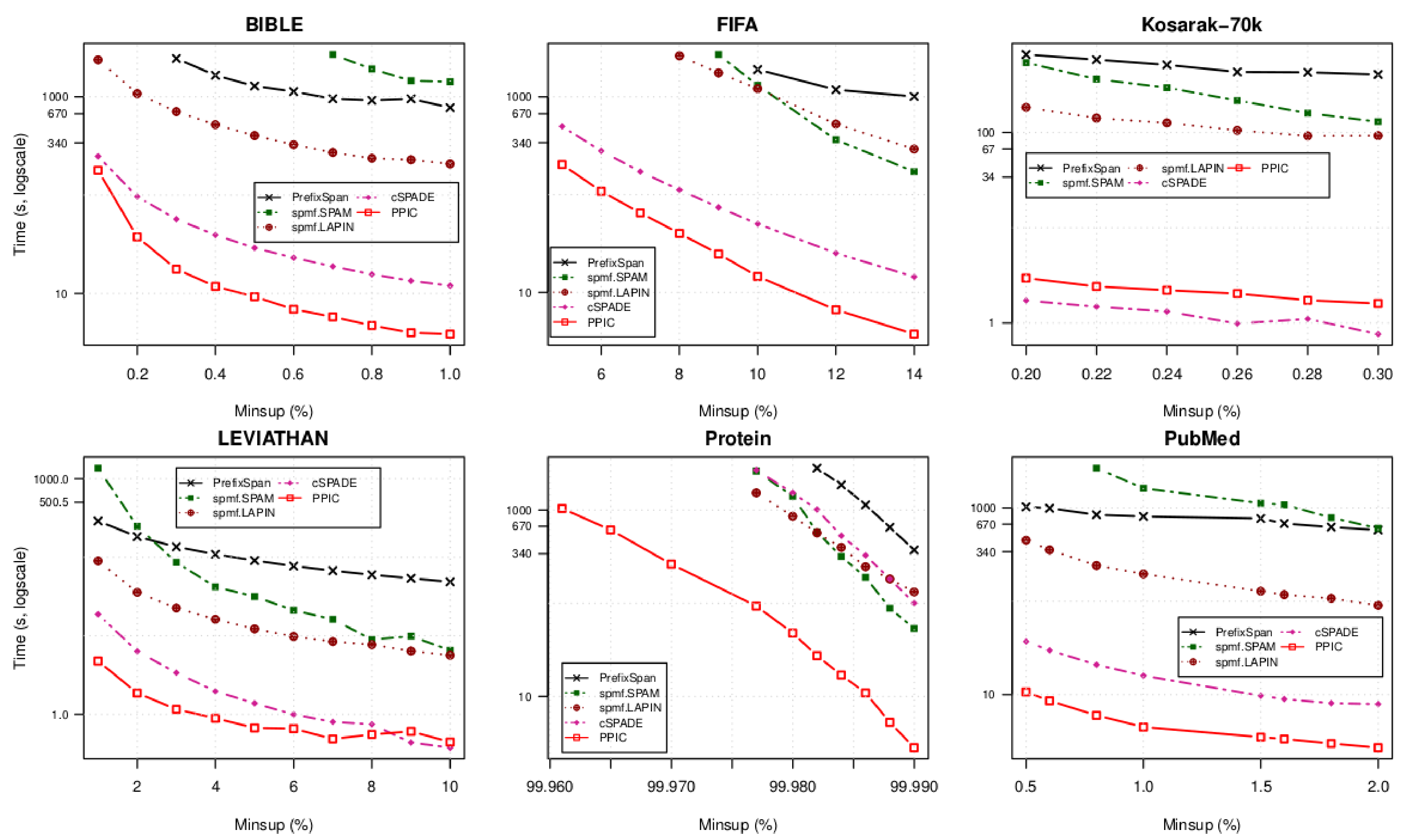
Extensions To SAT/MIP/CP Solvers

One global constraint for **all** sequences:

- algorithmic improvements: last position map, last position list
  $\rightarrow$ precomputed and cached, speedups
- use <u>backtracking-aware datastructure</u>
  $\rightarrow$ stores cover and prefix point in reversible vector

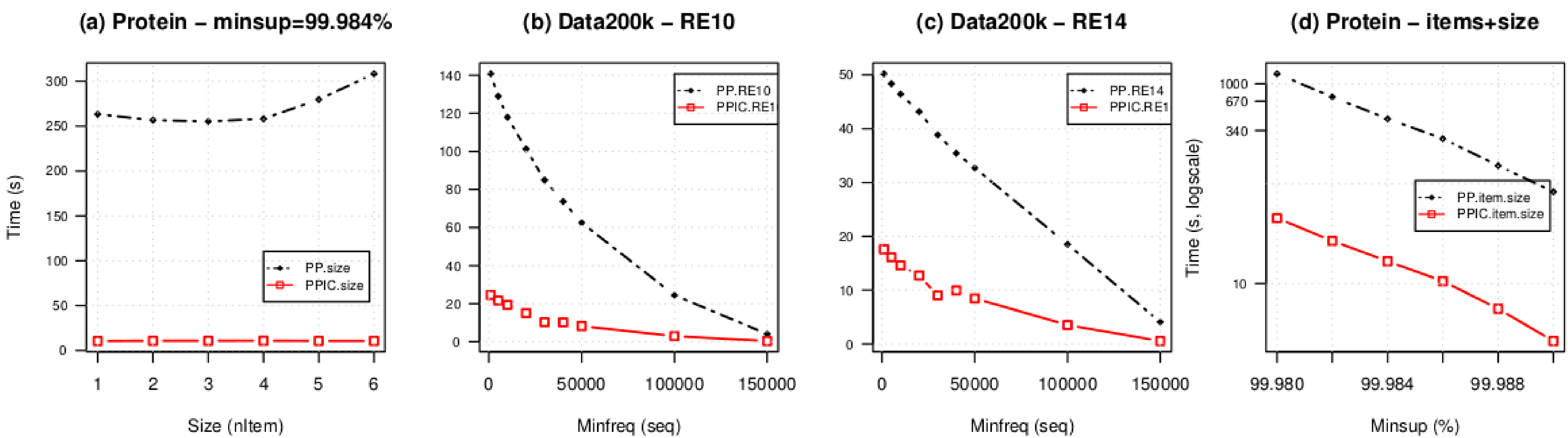


Efficiency: outperforms specialised!



[J. Aoga, T. Guns, P. Chaus, ECMLPKDD 2016]

Constraints



[J. Aoga, T. Guns, P. Chaus, ECMLPKDD 2016]

Improving generality

Constraints:

- Constraints on sequence: compatible
- Constraints on cover set: compatible
- Preferences over the solution set: compatible
- Constraints on inclusion relation: ???

=> best known: min/max gap and span

gap: 1 gap: 2

<Home, Work, Restaurant, Gym, Work, Home, Bar>

span: 4

can **modify the *global cover* constraint** to also enforce gap/span

$\rightarrow$ improves state-of-the-art (with backtrack-aware datastructure)

[J. Aoga, T. Guns, P. Chaus, CPAIOR 2017]

Other approaches

Classic Setting

- Encode cover as a regular expression [Coquery et al, ECAI 2012]
- Fixed-width sequences [Metivier et al, ECMLPKDD 2013]
- Formulation in pure ASP [Gebser et al, IJCAI 2016]
- Formulation in hybrid ASP [Paramonov, RuleML 2017]
- Episodes (in single sequence) [Cappart et al, CPAIOR 2018]

<u>Applications and novel settings that make use of CP:</u>

Extending Classic Settings

- informed gaps and bike-sharing analysis [Gay et al, ICMLA 2015]
- web-log mining [Kemmar et al, IJITW 2016]
- top-k and relevant sequence mining [Kemmar et al, Cons 2017]
- rare sequential patterns with ASP (care pathway mining) [Samet et al, ICLP 2017]

Sequence mining: modeling

Sequence mining: more complex *coverage* compared to itemsets

Classic Setting

Extending Classic Settings

Support for additional constraints:

- on syntax (inclusion/exclusion, regular expression, distance, ...):
  $\rightarrow$ for free
- on cover set/solution set:
  $\rightarrow$ some reusable from itemset mining
- on cover relation:
  $\rightarrow$ only by changing global constraint algorithm

Sequence mining: solving

Pure SAT/MIP/CP Formulation

Extensions To SAT/MIP/CP Solvers

Global constraint(s) for coverage:

- hides complexity
- fast algorithms inside (incremental, PrefixSpan-like)
- necessary for scalability
- even more efficient with backtracking-aware datastructures

OUTLINE

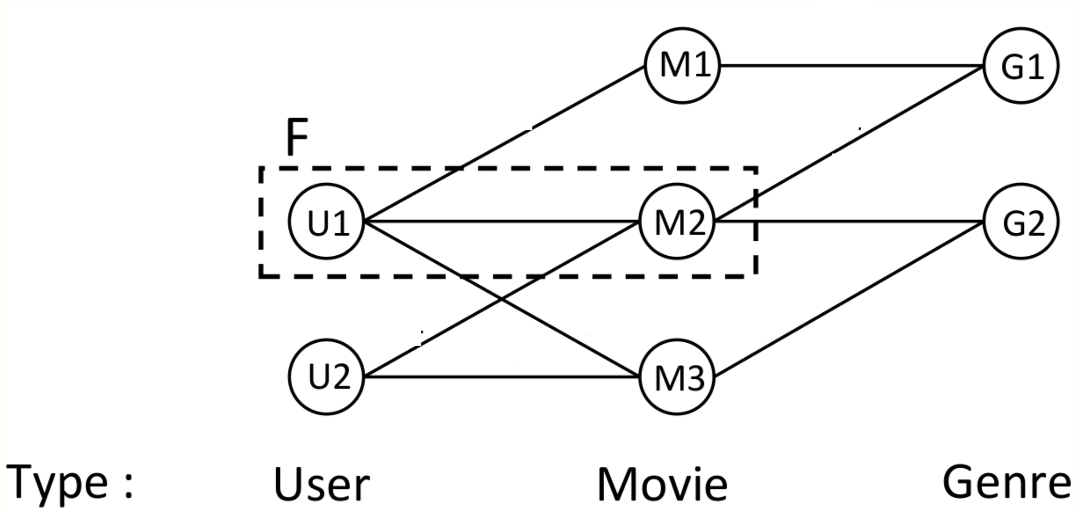
Previously
- itemsets
- pattern sets and decision trees

Up next: structured pattern mining!
- Sequence mining
- **Multi-relational patterns**
- *Interestingness* as objective function
- Wrap up of pattern mining

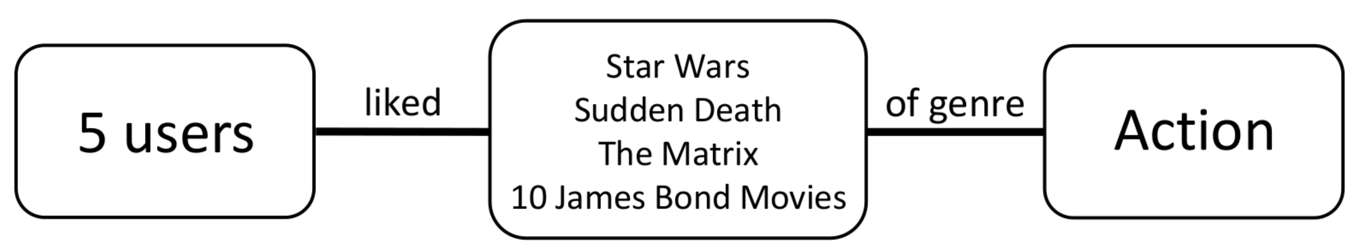
Example data and patterns

Itemset data: 2 entity types (items,transactions) and 1 relation

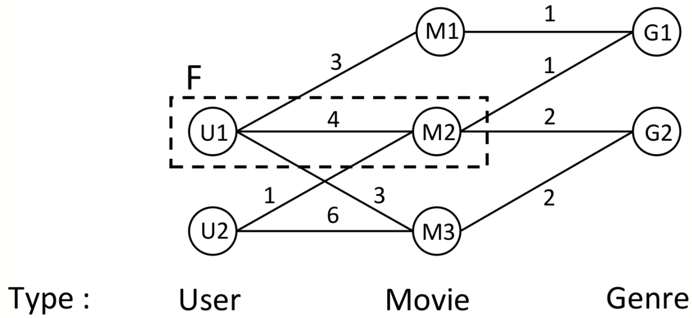
Multi-relational: example with 3 entity **types** and 2 relations



Example pattern from rottentomatoes dataset:



Why



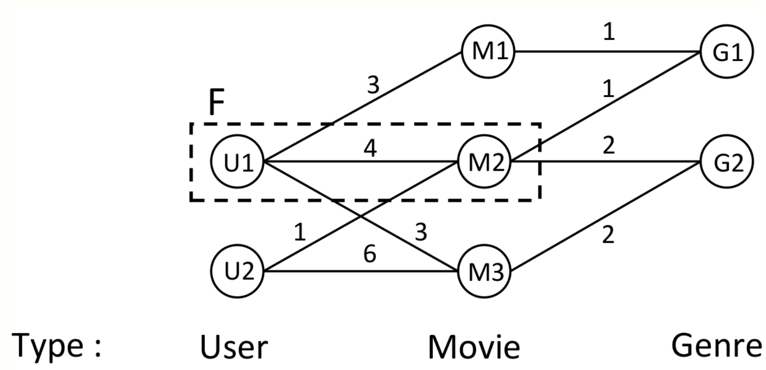
Data in the wild is often complex and **relational** (e.g. in databases)

Single relations only give limited view on interactions

Types of relations: chain (example above: chain of 3), star, triangle, ...

- generic techniques that work with any nr. of entities and relation between them
- many potential patterns, must define what a valid pattern is

Pattern definition



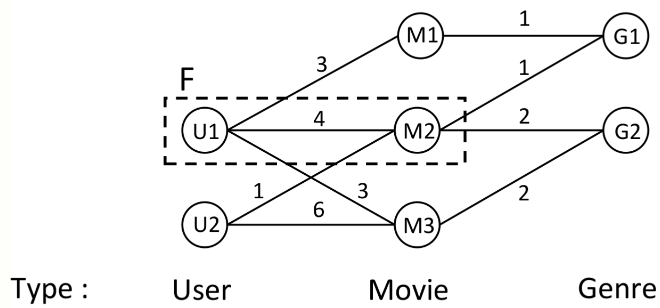
CCS pattern: Completely Connected Subset

- Subset: the <u>subset of entities</u> of each type defines the pattern
- Completely Connected: if U_1,M_2 in subset, <u>must be connected</u> in data

Generalizes <u>closed itemsets</u> to multiple relations

Constraint Programming model

Variables: one Boolean variable for each entity



value={0,1}: undecided

value=1: in

value=0: not in

→ *one solution to the CP = one CCS*

Constraints:

1) Completeness & maximal: very similar to itemsets
   → *can use similar global constraints, one per relation!*

2) Connected: there must be a path between all entities

Relational pattern mining

Pure SAT/MIP/CP Formulation

Writing an algorithm for relational pattern mining is <u>a lot of work</u>.

With CP:

- search and search heuristics for free
- can focus on implementing the individual constraints
  $\rightarrow$ natural decomposition of algorithm

Relational explosion

<u>Pattern explosion</u> even more pronounced:

- if one relation results in many patterns, two result in even more

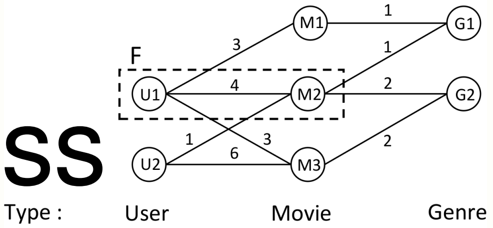
What are the most *interesting* patterns?

<u>Hot topic</u>:

subjective interestingness, e.g. through maximum entropy models

1) model prior knowledge of domain/user
   (from known popular items to known correlations)
2) find most statistically *surprising* pattern
3) update model of background knowledge, repeat

Patterns and interestingness



Subjective interestingness:

$$\frac{\sum_{e_1,e_2 \in F} -\log\left(p_{e_1,e_2}\right)}{a+b*|F|}$$

(self-information of edges)

(description length of nodes)

A scalable approach is missing: currently generate-and-rank,
but **generating all patterns often does not scale**

This is an optimisation problem, CP is made for both enumeration and optimisation

$\rightarrow$ difficult objective (real-valued, logarithms, fractional),
but can do advanced reasoning in a global objective constraint!

Extensions To SAT/MIP/CP Solvers

Bounds on the (non-linear) objective

$$\frac{\sum_{e_1,e_2 \in F} -\log(p_{e_1,e_2})}{a+b*|F|}$$

During search, each entity is *in*, *out* or *undecided*

Naive upper bound:

$$max\left(\sum_{e_1,e_2 \in F} p'_{e_1,e_2}\right) = \sum_{e_1,e_2 \in in} p'_{e_1,e_2} + all\ remaining\ edges$$

but not all remaining possible (unless clique)

$\rightarrow$ compute maximum nr. edges **based on degrees**

All denominators upper bound:

$$min\left(a+b*|F|\right) : try\ all\ possible\ |F| = \sum_t |e\ of\ type\ t|$$

all combinations of '***add n edges of type t***'

Add 'look-ahead' pruning:

– can derive when **adding an entity will worsen** objective

Experimental results

	#Rels	#Ent.	Dens.	#Sols.	LCM	RMiner	CP-closed	CP-naive	CP-denom	CP-prune
fimi/mushroom	1	8243	19.33%	238 709	4.55	timeout	110.5	10.3	4.89	2.33
fimi/chess	1	3271	49.33%	411 000 000+	timeout	timeout	timeout	16.4	16.2	14.9
fimi/T10I4D100K	1	100870	1.16%	359 000+	10.9	timeout	timeout	3915	468	87
fimi/T40I10D100K	1	100942	4.20%	1 000 000+	timeout	timeout	timeout	timeout	7643	224
fimi/connect	1	67686	33.33%	13 000 000+	timeout	timeout	timeout	2754	2097	1640
fimi/retail	1	104632	0.06%	3 000+	11.5	timeout	timeout	timeout	timeout	1421
foursquare/checkins	1	224947	2.11%	232 747	0.6	timeout	7296	255	35.5	16.8
imdb/1year	2	3291	0.97%	583	-	2128	1.05	1.26	0.4	0.1
imdb/5years	2	30131	0.24%	3 887	-	timeout	208	242	26.2	5.06
imdb/10years	2	51203	0.12%	8 704	-	timeout	1764	1618	127.9	8.36
imdb/40years	2	111320	0.03%	15 900+	-	timeout	timeout	timeout	990	9.18
imdb/100years	2	514323	0.002%	15 900+	-	timeout	timeout	timeout	timeout	290
rottentomatos	2	12263	0.44%	13 000 000+	-	timeout	timeout	timeout	2986	279
dblp-star	3	8279	0.10%	7 699	-	207	269	305	19	3
dblp-chain	3	13862	0.09%	30 629	-	76	5423	9141	939	107

[T. Guns, A. Aknin, J. Lijffijt, T. De Bie, ICDM 2016]

Relational pattern mining

Writing an algorithm for relational pattern mining is <u>a lot of work</u>.

With CP:

- search and search heuristics for free
- can focus on implementing the individual constraints
- can switch to optimisation easily

Here: added novel bounds for information-theoretic (and non-linear) objective function.

More relational settings possible, e.g. Triangle-Driven Community Detection in Large Graphs using SAT [Jabbour et al, 2018]

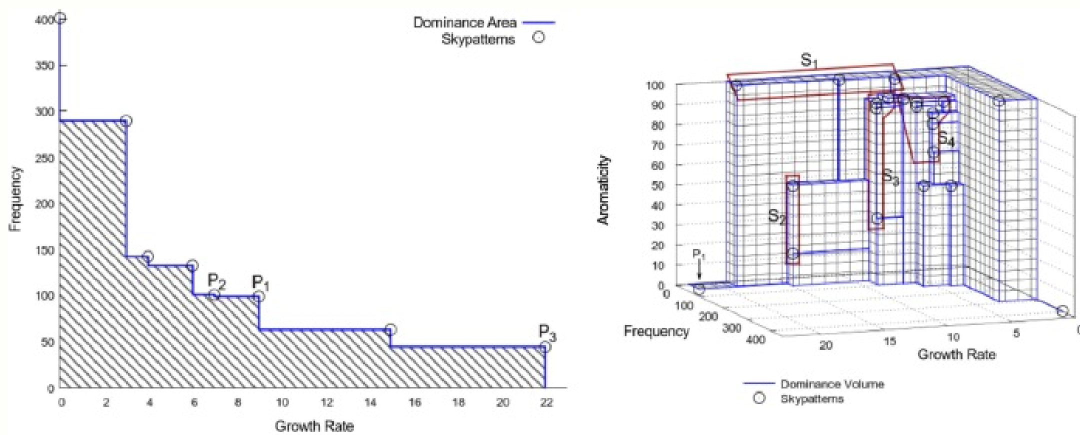
Interestingness measures and iterative mining

“Skypattern mining: From pattern condensed representations to dynamic constraint satisfaction problems”

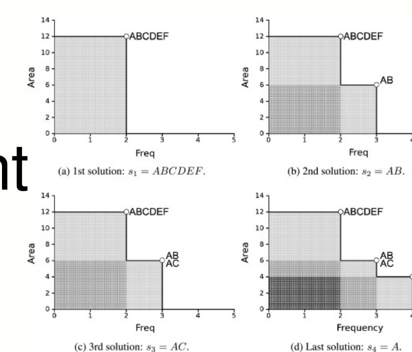
[W. Ugarte, P. Boizumault, B. Cremilleux, A. Lepailleur, S. Loudni, M. Plantevit, C. Raissi, A. Soulet, AIJ 2017]

Multi-objective optimisation

- frequency
- area
- growth-rate
- aromaticity
- ...



(a) Distribution of the skypatterns for M_1.
(b) Distribution of the skypatterns for M_4.



Uses CP to incrementally find a solution and add constraint
dynamic CSP

New settings

interestingness measures and iterative mining

"Flexible constrained sampling with guarantees for pattern mining"

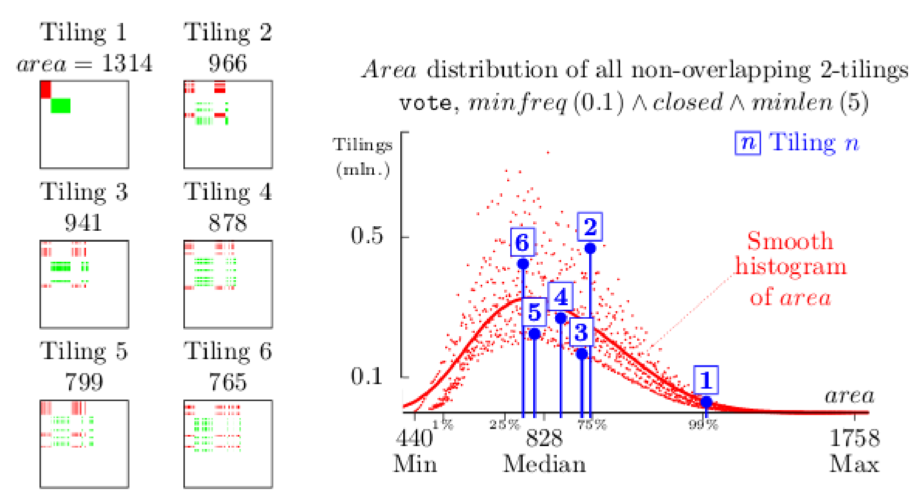
[V. Dzyuba, M. van Leeuwen, L. De Raedt, DMKD 2017]

1) sampling to avoid pattern explosion, proportional to qual. measure

2) multiple quality measures *and* constraints

Sampling with XOR constraints, [S. Chakraborty, D. Fremont, K. Meel, M. Vardi, AAAI 2014
on top of a CP-based itemset miner

Also for sampling pattern *sets*



OUTLINE

Previously

- itemsets
- pattern sets and decision trees

Up next: structured pattern mining!

- Sequence mining
- Multi-relational patterns
- *Interestingness* as objective function
- **Wrap up of pattern mining**

Structured pattern mining

Beyond sets:

- Multi-relational pattern mining
- Tree mining
- Graph mining
- Trajectory mining (spatio-temporal)
- ...

General structured pattern mining?

General structured pattern mining?

Abstraction of all pattern mining problems:

- Pattern type: *what is the structure of a pattern (set, seq, graph, ...)*
- Generating operator: *what are possible patterns (e.g. symbol set)*
  - includes canonicity operator: *only unique patterns (ismorphism)*
- Constraining operator: *what are valid patterns (e.g. seq is ordered)*
  - includes matching operator: *when is a transaction covered*

Different ways to model declaratively or implement, or hybrid...

[On declarative modeling of structured pattern mining, DeBLP @ AAAI, 2016]

Pattern Mining: why CP?

For modeling:

- rapid prototyping
- variations of existing problems (reuse constraints)
- as part of bigger combinatorial problem (novel settings)

For solving:

- reuse of existing solvers
- for increased efficiency: reuse or develop global constraints
- hybredize (at its core, CP is just a Depth-First Search engine)

Pattern Mining: when CP?

Efficiency vs generality trade-off

- Does the problem have many constraints?
- Does the problem benefit from a generic approach?
- How important is efficiency?
- Can you reuse existing global constraints?

Itemsets: many constraints, all possible with existing constraints

Sequences: many constraints, some reuse (freq, regex) some custom

Graphs: no constraints immediately reusable...

Pattern mining in CP

Future challenges:

- Efficiency vs generality trade-off
  (ex. global constraints and their granularity)
- DFS framework: branch-and-bound and novel bounds
- Pattern set mining:
  - iterative methods (iterative branch-and-bound)
  - (constrained) sampling framework
  - hybrid methods
- Novel and complex problem settings

Structured mining, references 1

Coquery, Emmanuel, Said Jabbour, Lakhdar Sais, and Yakoub Salhi. "A SAT-based approach for discovering frequent, closed and maximal patterns in a sequence." In Proceedings of the 20th European Conference on Artificial Intelligence, pp. 258-263. IOS Press, 2012.

Métivier, Jean-Philippe, Samir Loudni, and Thierry Charnois. "A constraint programming approach for mining sequential patterns in a sequence database." ECMLPKDD 2013

Kemmar, Amina, Willy Ugarte, Samir Loudni, Thierry Charnois, Yahia Lebbah, Patrice Boizumault, and Bruno Crémilleux. "Mining relevant sequence patterns with cp-based framework." In Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on, pp. 552-559. IEEE, 2014.

Gay, Pablo, Beatriz López, and Joaquim Meléndez. "Learning Complex Events from Sequences with Informed Gaps." In Machine Learning and Applications (ICMLA), 2015 IEEE 14th International Conference on, pp. 1089-1094. IEEE, 2015.

Kemmar, Amina, Samir Loudni, Yahia Lebbah, Patrice Boizumault, and Thierry Charnois. "PREFIX-PROJECTION global constraint for sequential pattern mining." In International Conference on Principles and Practice of Constraint Programming, pp. 226-243. Springer, 2015.

Kemmar, Amina, Samir Loudni, Yahia Lebbah, Patrice Boizumault, and Thierry Charnois. "A global constraint for mining sequential patterns with gap constraint." In International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems, pp. 198-215. Springer International Publishing, 2016.

Structured mining, refs 2

Aoga, John OR, Tias Guns, and Pierre Schaus. "An efficient algorithm for mining frequent sequence with constraint programming." In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 315-330. Springer International Publishing, 2016.

Gebser, Martin, Thomas Guyet, René Quiniou, Javier Romero, and Torsten Schaub. "Knowledge-based sequence mining with ASP." In IJCAI 2016-25th International joint conference on artificial intelligence, p. 8. AAAI, 2016.

Kemmar, Amina, Yahia Lebbah, and Samir Loudni. "A Constraint Programming Approach for Web Log Mining." International Journal of Information Technology and Web Engineering (IJITWE) 11, no. 4 (2016): 24-42.

Aoga, John OR, Tias Guns, and Pierre Schaus. "Mining time-constrained sequential patterns with constraint programming." Constraints (2017): 1-23.

Kemmar, Amina, Yahia Lebbah, Samir Loudni, Patrice Boizumault, and Thierry Charnois. "Prefix-projection global constraint and top-k approach for sequential pattern mining." Constraints 22, no. 2 (2017): 265-306.

Samet, Ahmed, Thomas Guyet, and Benjamin Negrevergne. "Mining rare sequential patterns with ASP." In International Conference on Inductive Logic Programming. 2017.

Paramonov, Sergey, Stepanova Daria, and Miettinen Pauli. "Hybrid ASP-based approach to pattern mining." In RuleML+ RR conference proceedings. Springer, 2017.

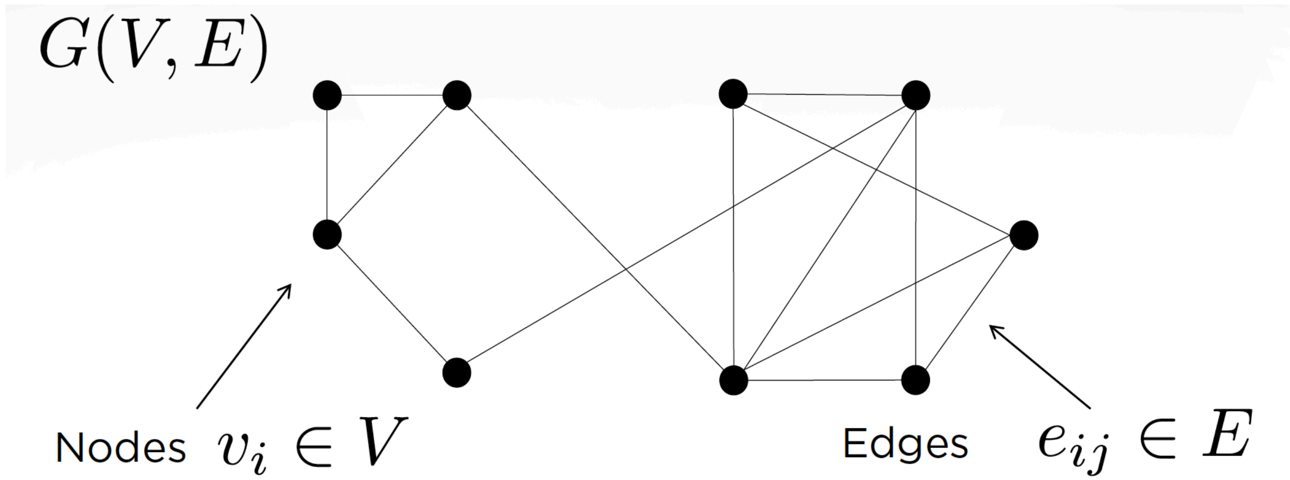
Clustering Module

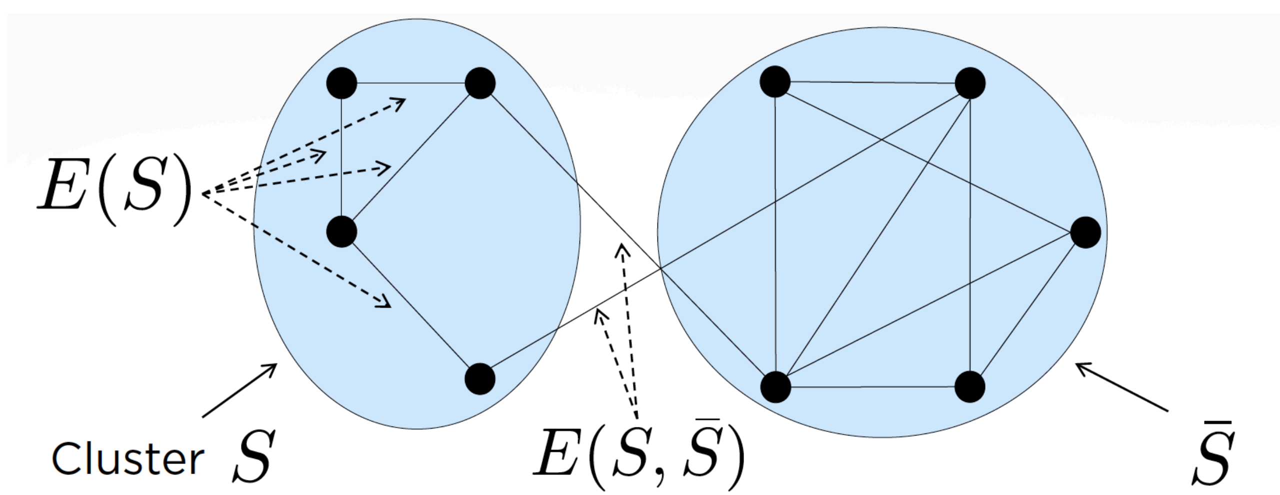
What is Clustering

- Given a set of instances, *S,* i.e. images, documents, DB records, nodes in a graph etc.
- Simplest form, find the "best" *k* block set partition

  $S = \{S_1 \cup S_2 \cup \ldots S_k\}$
- Multitude of algorithms that define "best" differently
  - K-Means, K-Medians, K-Centers
  - Spectral Graph Clustering, Mixture Models
  - Self Organized Maps
- Aim is to find the **underlying** structure/patterns/ groups in the data:
  - For Emails -> Clusters = topics
  - Pixels in an Image -> Clusters = objects
  - Graphs -> Clusters = communities

Illustrative Example – Social Networks

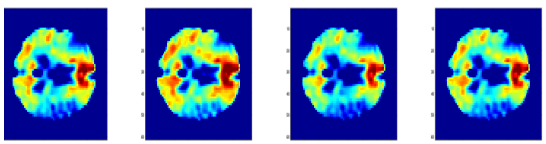


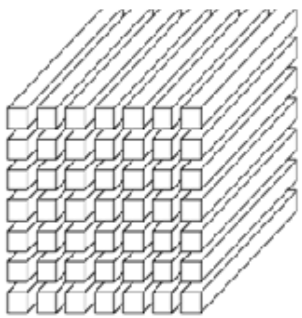


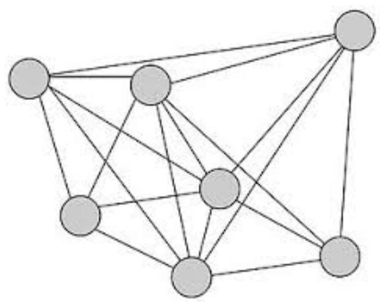
Lots of Real World Problems are Naturally Clustering – Medical Imaging

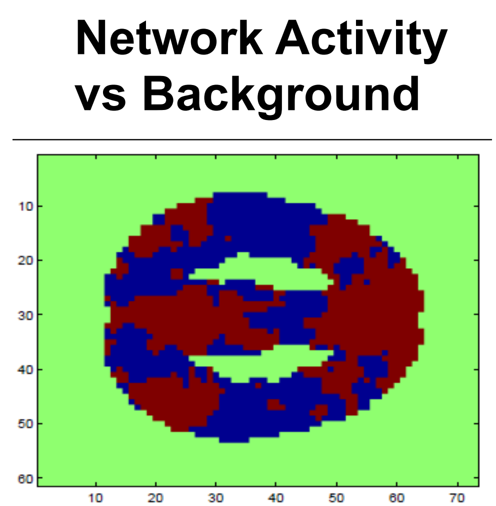
[With NMRC, Pennington Institute, UC Davis Medical Imaging Group]

Can We Discover What Cognitive Networks Are Associated With Tasks?





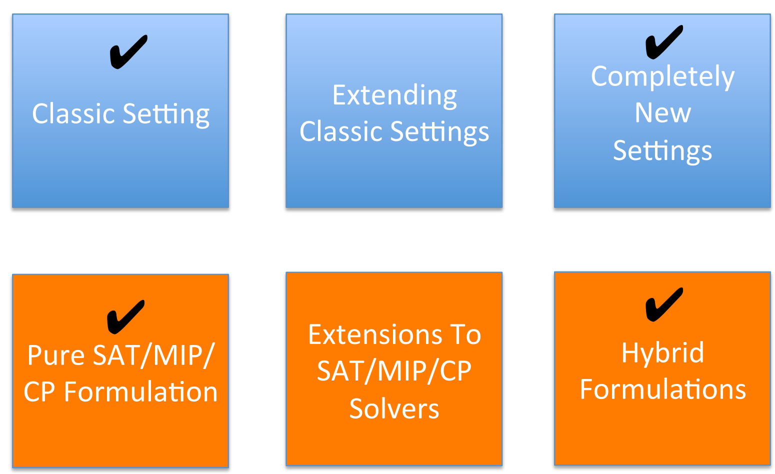




Plan For The Clustering Module

- Adding **explanation** to clustering
  - How can we make clustering explainable
- **Classic** formulations of clustering
  - New insights into existing problems
- Using CP to add **constraints** to clustering
  - Changing solvers to be more efficient for ML
- **Human in the loop** style minimal modification clustering
  - A new setting which allows constraints to be a mechanism for dialog b/w humans and machines

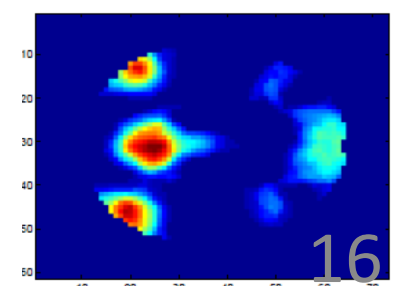
Adding Explanation to Clustering



Motivating the Need For Explanation - Precision Medicine for SZ Treatment

- Current Approach for SZ Treatment Recommendation
  - Your starting to become delusional
  - Interview with a doctor
  - Recommends (**68% effective**) a long treatment plan
- Precision Medicine Approach
  - Place a person in a MRI machine
  - Learnt what brain network activity is associated with treatment plan successes (i.e. biomarkers in neural activity) (**83% effective**)

Joint work with U.C. Davis Imaging Center



Precision Medicine Application for SZ Treatment Prediction

- Not that easy, simply stating the method is better than humans is insufficient.
- Need for:
  - Explain that the results are **fair** to the FDA (or equivalent).
    - Are we giving expensive treatments disproportionately to some groups?
  - Detailed explanation to our collaborators to determine neurologically plausible.
    - Reduced connectivity between ROI x and y
  - High level explanations to build **trust** by other practitioners (i.e. best practices committee at HMO).
    - Dampening of connectivity in Executive functional network

Explanation Is Not Always Needed: Sign Recognition for Google Car

- My student Aubrey Gress spent a summer working at Google so the next driverless car can read signs.

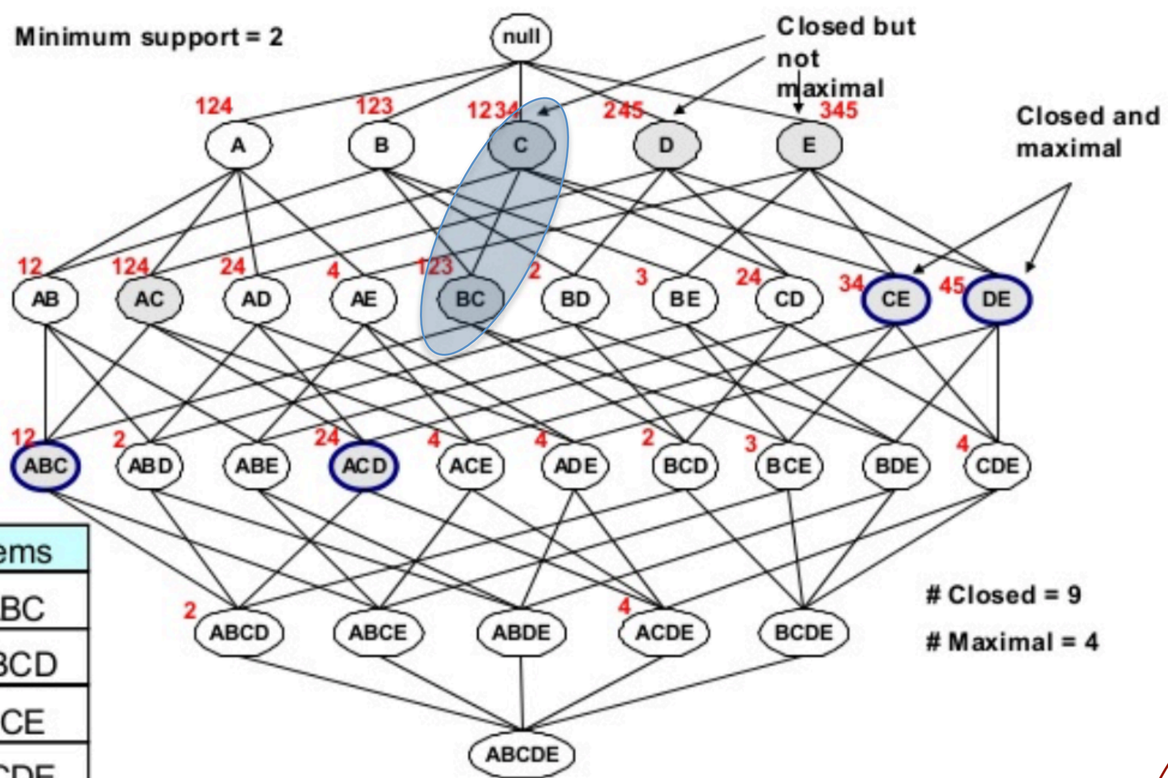
The criteria for success is accuracy
i.e. If recognizing STOP sign by outer white band is better than recognizing the letters "STOP" so be it.

Adding Explanation/Description To Clustering

- Three ways:
  - **Before**: Find concepts in the data, build the clustering around the concepts − Universität Mainz @ DS 10, U. Caen @ IJCAI 16
  - **After**: Take an existing clustering, explain what differentiates each cluster from the others. − Davis & Lyon @ NIPS 18
  - **During:** Simultaneously find a clustering that is also explainable − Davis & Orleans @ IJCAI 18

Closed Itemsets as Concepts

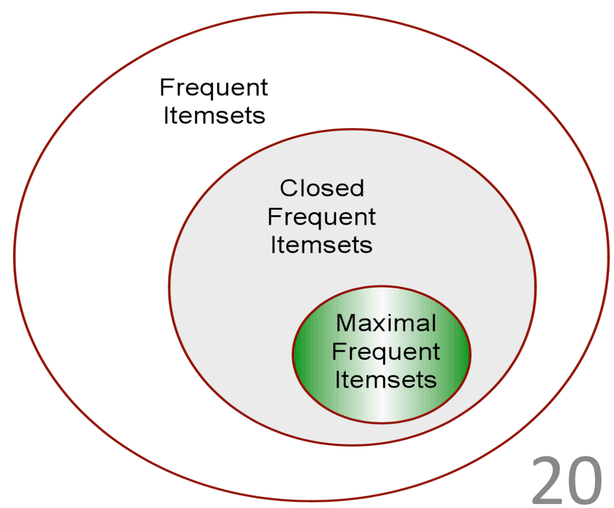
[Ouali, A., Loudni, S., Lebbah, Y., Boizumault, P., Zimmermann, A., and Loukil, L. Efficiently finding conceptual clustering models with integer linear programming. IJCAI'16]



TID	Items
1	ABC
2	ABCD
3	BCE
4	ACDE
5	DE

Closed Itemsets

C
D
E
AC
BC
CE
DE
ABC
ACD



Concept = CI

Multiple concepts per Instance

Can't use all concepts, choose a subset

Combining With Existing Algorithms

[Ouali, A., Loudni, S., Lebbah, Y., Boizumault, P., Zimmermann, A., and Loukil, L. Efficiently finding conceptual clustering models with integer linear programming. IJCAI'16]

- Step 1: Computes closed item sets using LCM (or something else).
- Q: Which CI to use as concepts?
- Bi partite graph view

Is CI c chosen?

Some measure of interest associated with CI c

Optimize	$\sum_{c \in \mathcal{C}} v_c \cdot x_c$
Subject to	(1) $\sum_{c \in \mathcal{C}} a_{t,c} \cdot x_c = 1, \quad \forall t \in \mathcal{T}$ (2) $\sum_{c \in \mathcal{C}} x_c = k_0$ $x_c \in \{0,1\}, c \in \mathcal{C}$

Every instance must have one concept

Combining With Existing Algorithms

[Ouali, A., Loudni, S., Lebbah, Y., Boizumault, P., Zimmermann, A., and Loukil, L. Efficiently finding conceptual clustering models with integer linear programming. IJCAI'16]

- Diversity extension

Some measure of interest associated with CI c

Is CI c chosen?

Optimize	$\sum_{c \in \mathcal{C}} v_c \, . \, x_c$
Subject to	(1) $\sum_{c \in \mathcal{C}} a_{t,c} \, . \, x_c = 1, \quad \forall t \in \mathcal{T}$
	(2) $k = \sum_{c \in \mathcal{C}} x_c$
	(2′) $k_{min} \leq k \leq k_{max}$
	(3) $\sum_{c \in \mathcal{C}} w_{i,c} \, . \, x_c = 1, \quad \forall i \in \mathcal{I}$
	$k \in \mathbb{N}, \quad x_c \in \{0,1\}, c \in \mathcal{C}$

Does transaction t contain CI c

Does item i belongs to the closed itemset c

Runtime Analysis Based on CI Size

Ouali, A., Loudni, S., Lebbah, Y., Boizumault, P., Zimmermann, A., and Loukil, L. Efficiently finding conceptual clustering models with integer linear programming. IJCAI'16

Can **scale** method by increasing minimum support

dataset	#transactions	#items	density(%)	Number of closed itemsets	CPU Time (s.) (1)	CPU Time (s.) (2)
Soybean	630	50	32	31,759	0.74	15.52
Primary-tumor	336	31	48	87,230	1.98	40.62
Lymph	148	68	40	154,220	3.38	25
Vote	435	48	33	227,031	4.42	191.12
tic-tac-toe	958	27	33	42,711	0.07	6.22
Mushroom	8124	119	18	221,524	17.1	154.75
Zoo-1	101	36	44	4,567	0.38	420.75
Hepatitis	137	68	50	3,788,341	236.85	1,493.38
Anneal	812	93	45	1,805,193	113.84	20,825.8

Table 3: Dataset characteristics.

Pre-processing step (LCM)

Solve ILP

Constraints on Clusters Combinations

Mueller, Marianne, and Stefan Kramer. "Integer Linear Programming Models for Constrained Clustering." Discovery science 2010. Vol. 6332. 2010.

Overlapping formulation

instance i covered by y patterns

Upper bound on number of overlaps

$$\begin{array}{lll}
\text{maximize} & \frac{1}{k}(w_{max} - w)^T x & \\
\hline
\text{subject to} & \text{(i) } Ax = y & \text{(vi) } \mathbf{1}^T v \leq n \cdot maxOverlap \\
& \text{(ii) } \mathbf{1}^T x = k & \text{(vii) } x \in \{0,1\}^n \\
& \text{(iii) } z \geq \mathbf{1} - y & \text{(viii) } y \in \mathbb{N}_0^m \\
& \text{(iv) } \mathbf{1}^T z \leq (m - m \cdot minCompl) & \text{(ix) } v \in \mathbb{N}_0^m \\
& \text{(v) } v \geq y - \mathbf{1} & \text{(x) } z \in \{0,1\}^m
\end{array}$$

Scales quite well if minsupp is high enough

Set to 0 if instance covered

Some Unanswered Questions

- Bounds on k to ensure (1) can be satisfied: that there exists a feasible solution?
- Other complex constraints involving ontologies
  - $x_i \vee x_j \Rightarrow x_k$: if the concept of pulmonologist or cardiologist is used need to use physician

Optimize	$\sum_{c \in \mathcal{C}} v_c \cdot x_c$
Subject to	(1) $\sum_{c \in \mathcal{C}} a_{t,c} \cdot x_c = 1, \quad \forall t \in \mathcal{T}$ (2) $\sum_{c \in \mathcal{C}} x_c = k_0$ $x_c \in \{0, 1\}, c \in \mathcal{C}$

Cluster Using One Set of Features Explain Using Another?

[The Cluster Explanation Problem: Complexity Results, Algorithms and Applications, Davidson et al. Neurips 18]

- Why?
- Features used for clustering are not interpretable
  - Deep learning representations (Word2Vec, Bert, AE)
  - Graphs i.e. social networks
- Features for clustering are private/sensitive
- Features used to obtain clustering are no longer available (historical clusters)
  - Electoral district maps

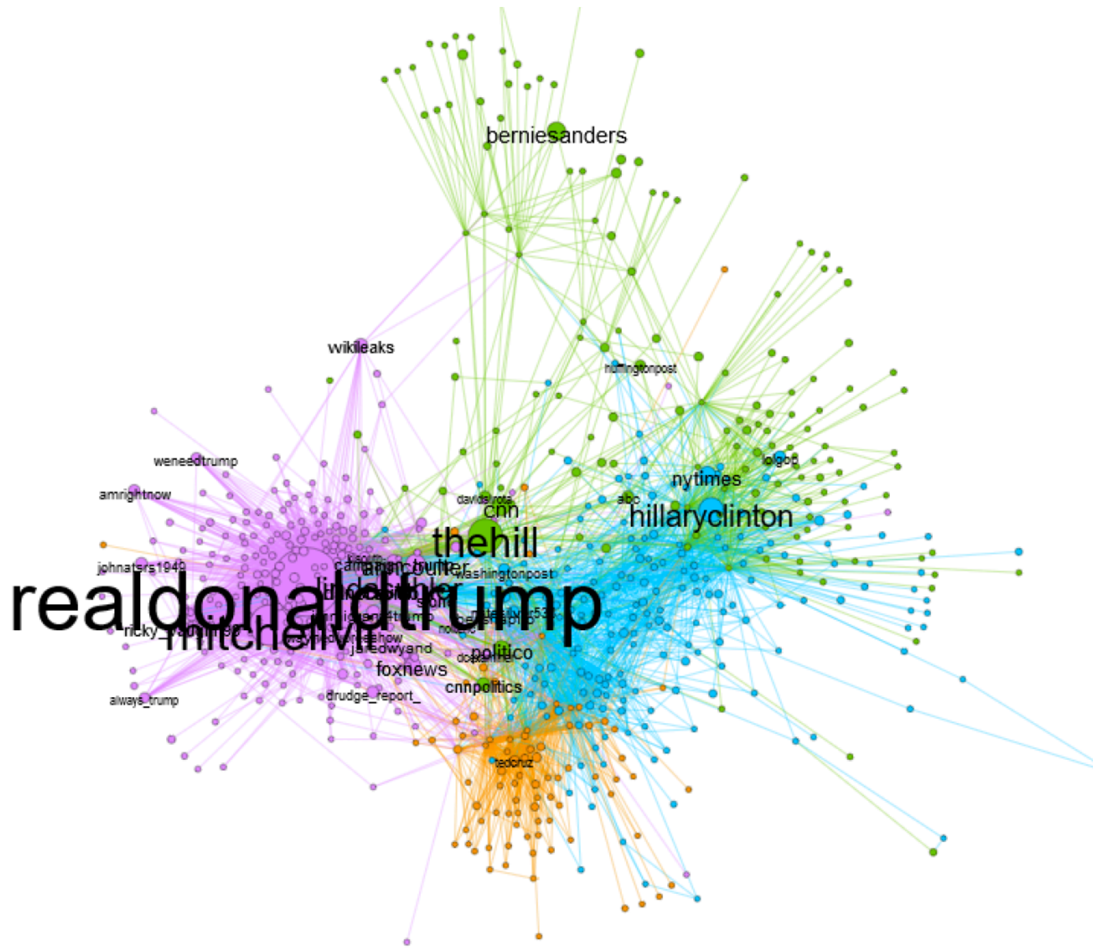
Twitter Data from ERIC Lab Univeristy Lyon - 2

- Election Tweets (**French and USA**) from 01/01/2016 to 22/08/2016
  - Covers the primary season for the USA
- Communities formed based on **follower network**
- Explain communities using **hashtag usage**

HASHTAG

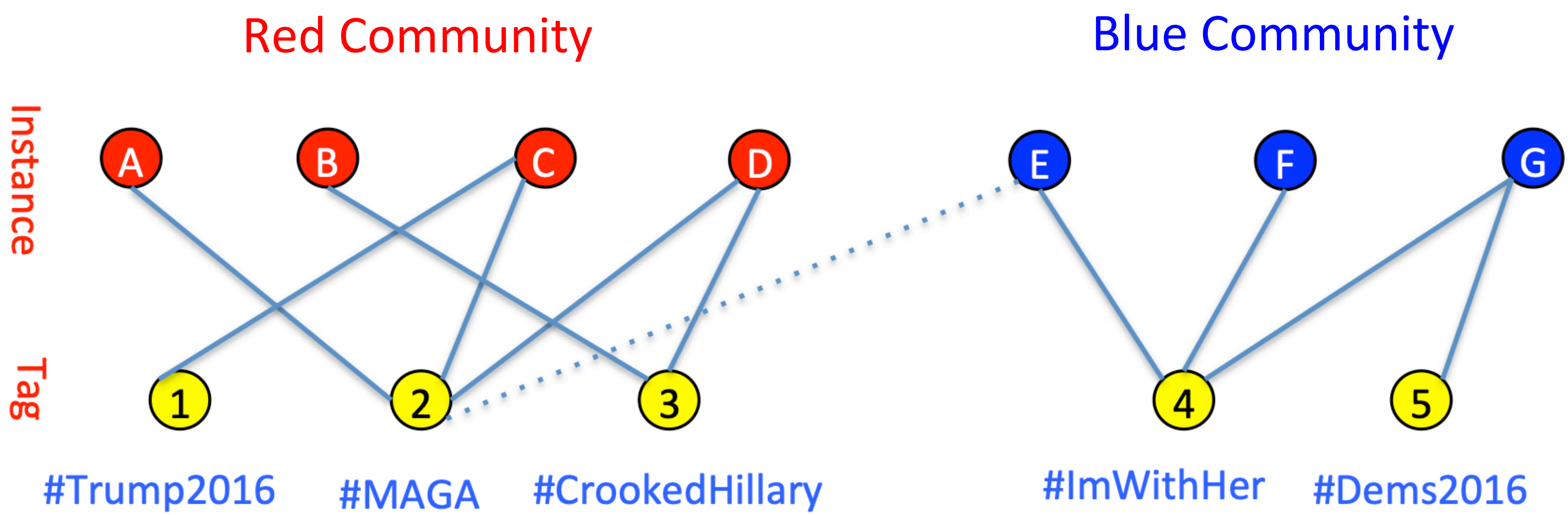
Segmenting Graphs

- Many classic algorithms – i.e. Louvain method
- Finds useful communities
- But doesn't explain what they are talking about.



Modeling the Explanation Problem as a Bipartite Graph

A Simple Example with Just Two Clusters



Explanation Problem (informally): Pick the smallest subset of yellow nodes/tags that “cover” all red instances. Chose a **different** subset of yellow nodes/tags for the blue tags

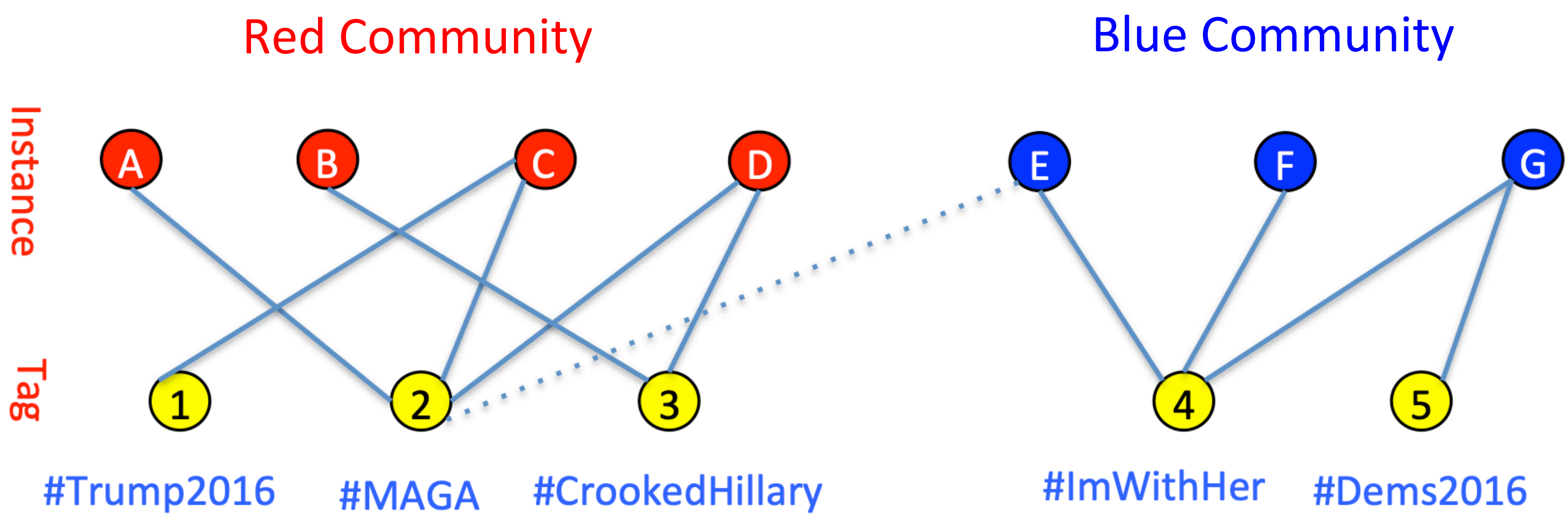
Formalizing DTDF Problem

[Disjoint Tag Descriptor Feasibility Problem]

The goal is to find a subset $T_j \subseteq T$ of tags for each cluster C_j $(1 \leq j \leq k)$ such that all the following conditions are satisfied.

(a) For each cluster C_j and each item $s_i \in C_j$, T_j has at least one of the tags in t_i; formally, $|T_j \cap t_i| \geq 1$, for each $s_i \in C_j$ and $1 \leq j \leq k$.

(b) The sets $T_1, T_2, \ldots, T_k$ are pairwise disjoint.



Formalizing This As An ILP

A triple set to 1 iff instance *i* is in cluster *k* and uses tag *j*

Find a simple explanation

$$argmin_X \sum_{i,j} X_{i,j}$$

Cover each and every instance

$$s.t. \quad \sum_j X_{k,j} S^k_{i,j} \;\geq\; 1 \;\; \forall\, i \in C_k,\; \forall\, k$$

Upper bound overlap

$$s.t. \quad \sum_i X_{i,j} \;\leq\; w_j \;\;\; \forall\, j$$

This is very similar to the set cover problem, one of Karp's 21 original intractable problems.

Formalizing This As An ILP

$$argmin_X \sum_{i,j} X_{i,j}$$

Find a simple explanation

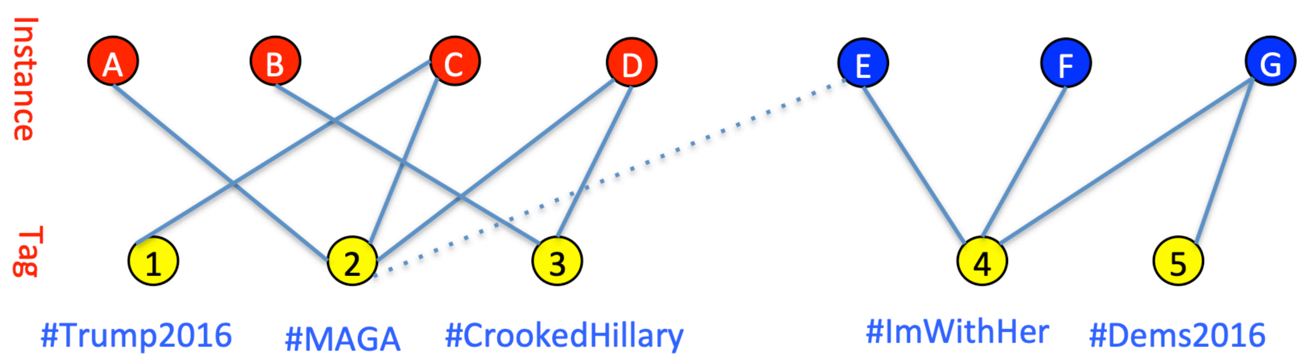
$$s.t. \quad \sum_j X_{k,j} S^k_{i,j} \; \geq \; 1 \;\; \forall \; i \in C_k, \; \forall \; k$$

Cover each and every instance

$$s.t. \quad \sum_i X_{i,j} \; \leq \; w_j \;\; \forall \; j$$

Upper bound overlap

Solution?



Formalizing This As An ILP

$$argmin_X \sum_{i,j} X_{i,j}$$

Find a simple explanation

$$s.t. \quad \sum_{j} X_{k,j} S^{k}_{i,j} \;\geq\; 1 \quad \forall\, i \in C_k,\; \forall\, k$$

Cover each and every instance

$$s.t. \quad \sum_{i} X_{i,j} \;\leq\; w_j \quad \forall\, j$$

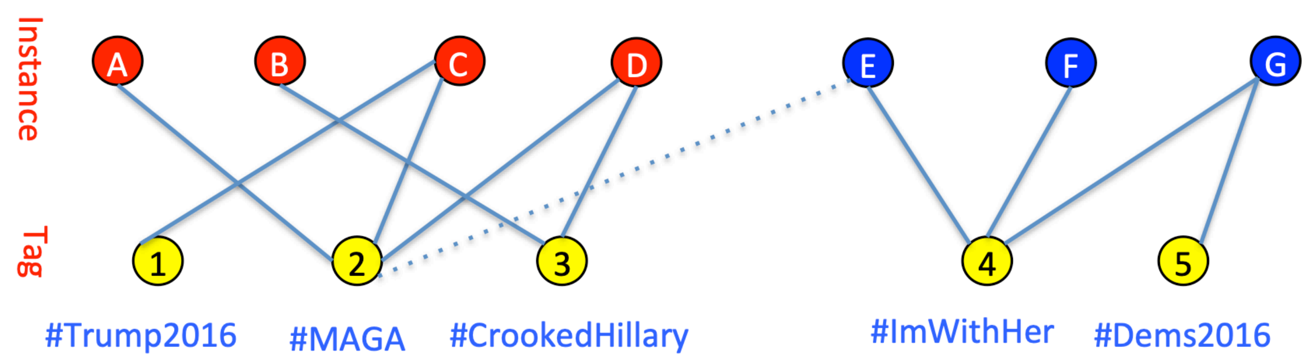
Upper bound overlap

Optimal Solution

X_R = {MAGA, CrookedHillary}

X_B = {ImWithHer}

But if E only has the MAGA tag, no feasible Solution exists



Formalizing This As An ILP

$$argmin_X \sum_{i,j} X_{i,j}$$

Find a simple explanation

$$s.t. \quad \sum_j X_{k,j} S^k_{i,j} \; \geq \; 1 \;\; \forall \; i \in C_k, \;\; \forall \; k$$

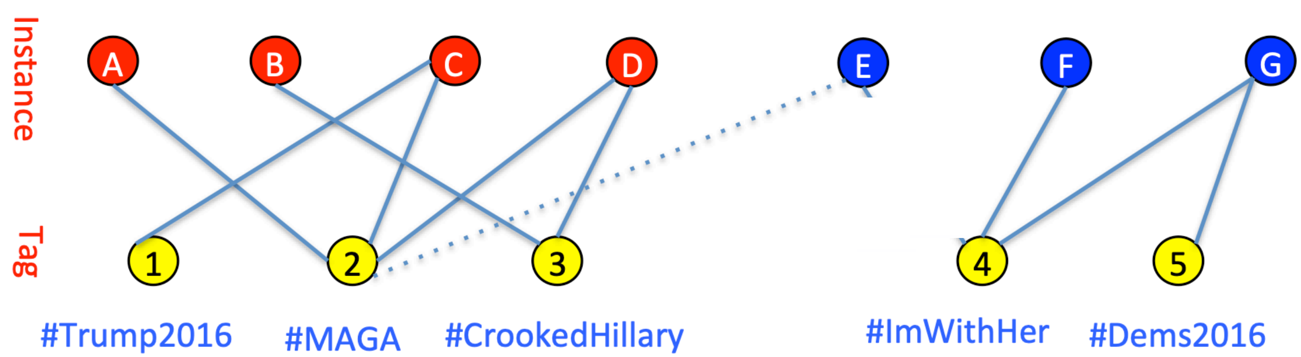
Cover each and every instance

$$s.t. \quad \sum_i X_{i,j} \; \leq \; w_j \;\; \forall \; j$$

Upper bound overlap

Optimal Solution?

But if E only has the MAGA tag, no feasible Solution exists



Variation #1

- Cover or forget (constraint replacement)

$$s.t.\ \ z_i + \sum_j X_{k,j} S^k_{i,j} \geq 1 \ \forall\, i \in C_k,\ \forall\, k$$

$$s.\ \ \sum_i z_i \ \leq \ I_k \ \ \forall i \in C_k,\ \forall\, k$$

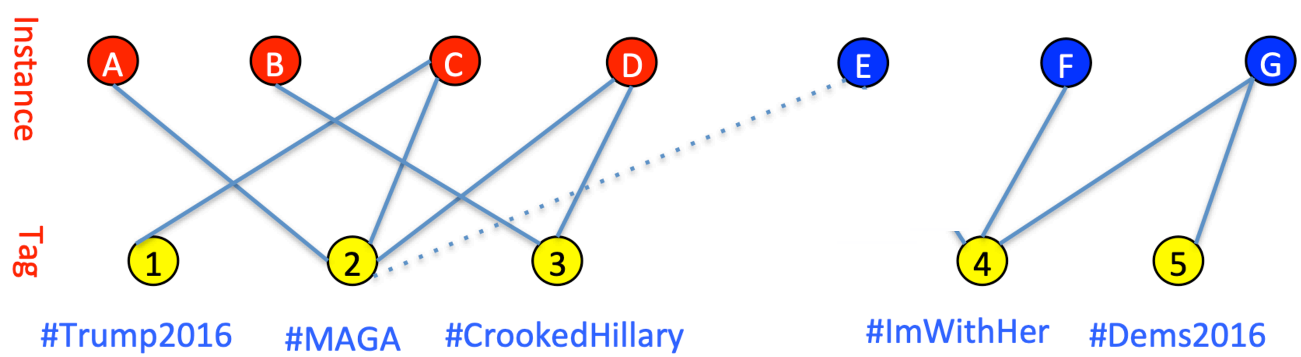
Optimal Solution

X_R = {MAGA, CrookedHillary}

X_B = {ImWithHer}

Z_5 = 1

If E only has the MAGA tag



Variation #2

- Composition constraints

$$s.t.\ \ X_{k,i} + X_{k,j} \leq 1\ \ \forall\ \{i,j\} \in \texttt{Apart},\ \forall\ k$$ (#MAGA, #Clinton)

$$s.t.\ \ X_{k,i} = 1 \rightarrow X_{k,j} = 1\ \ \forall\{i,j\} \in \texttt{Together},\ \forall\ k$$ (#MAGA, #Trump)

Lemma 1

Any disjunction of literals $v_1 \ldots v_m$ can be represented by a linear inequality (i.e. $v_1 \vee v_2 \ldots \vee v_m \equiv \sum_i \mathbf{v}_i \geq -m + 2$).

Lemma 2

Any conjunction of literals $v_1 \ldots v_m$ can be represented by a linear equality (i.e. $v_1 \wedge v_2 \ldots \wedge v_m \equiv \sum_i \mathbf{v_i} = m$).

Theorem 3

Given a set of literals $v_1 \ldots v_m$, any set of clauses using those literals in conjunctive normal form can be represented by a system of linear inequalities: $A_{=}\mathbf{v} = \mathbf{b}_{=}, A_{\geq}\mathbf{v} \geq \mathbf{b}_{\geq}$.

TrumpTrain ⋁ MAGA =>
Not(ImWithHer ⋁ Clinton)

AAAI 13 Paper of Ours

Contributions

Theorem 3.1 *The* DTDF *problem is* ***NP****-complete even when the number of clusters is 2 and the tag set of each item is of size at most 3.*

What were the three options if the problem is intractable?.

Contributions

Theorem 3.1 *The* DTDF *problem is* ***NP****-complete even when the number of clusters is 2 and the tag set of each item is of size at most 3.*

What were the three options if the problem is intractable?

If we require:

i) each explanation must have at most α tags

ii) no two explanations may have more than β tags in common.

Theorem 5.1 *The* (α, β)-CONS-DESC *problem can be solved in polynomial time when the number of clusters* k *is fixed. This algorithm can also handle* `Together` *and* `Apart` *composition constraints.*

This is called a fixed parameter tractable problem

Simple Fixed Parameter Tractable Algorithm

Algorithm 1: Description of our Algorithm for (α, β)-CONS-DESC

Input : A collection of k clusters $C_1, C_2, \ldots, C_k$ with tag sets for each instance in each cluster.
Output : A valid descriptor with at most α tags for each cluster such that any pair of descriptors have at most β tags in common. (Please see the main text for the definition of a valid descriptor.)

for Cluster C_1 **do**
  Get the next valid descriptor D_1.
  for Cluster C_2 **do**
    Get the next valid descriptor D_2.
      $\vdots$
    for Cluster C_k **do**
      Get the next valid descriptor D_k.
      Let $\mathcal{D} = (D_1, D_2, \ldots, D_k)$.
      if *Each pair of descriptors in $\mathcal{D}$ have at most β tags in common* **then**
        Output $\mathcal{D}$ as the solution and **stop**.
      end
    end
  end
end
Print "No solution".

Let N be the most tags used in a cluster

Possible Explanations N Choose α = $O(N^{\alpha k})$

$|T_i| \leq \alpha$ hence $O(\alpha^2)$ to check constraints
Trivial to check for βoverlap

Code and Future Work

- https://web.cs.ucdavis.edu/~davidson/description-clustering/
  - See readme file
- The algorithm is polynomial, but is just brute force search
  - Can we use a clever branch and bound method
- Meta information about the tags
- Lots of other types of explanations beyond disjunctions
  - CNF, DNF
- Explanations using other types beyond tags
  - Need to use SMT, OMT solvers not ILP solvers
- Measuring stability of explanations for measures of trust

Again This Setting – Simultaneously find Clustering and Explanation

[Descriptive Clustering: ILP and CP Formulations with Applications, Dao1 et al. IJCAI 2018]

Setting	Features	Descriptors/Tags
Twitter network	Mention/Retweet Graph	Hashtag usage
Images	SIFT features	Caption
EHR	Health record	Symptoms

- Many domains have a set of very good features/ attributes to form compact clusters on
  - Graphs, SIFT features for images,
- But they cannot **explain** the clustering well
- Instead we have another set of (potentially **sparse and noisy**) descriptors that are useful for explanation

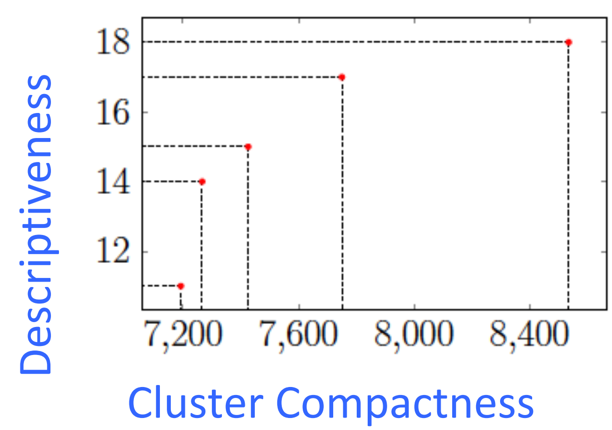
Descriptive Clustering Problem

- Two objectives find: f(s) a good clustering (using the features) and g(s) a **useful** description (using descriptors)
- Objectives need not be **compatible**
- Natural trade off between minimizing f(s) and maximizing g(s).
    - Compact clusters have too simple descriptions
    - Wide clusters have too complex descriptions
- ML has a tradition of adding together objective functions with a hyper-parameter to tune.
    - i.e. Model Fit and Model Complexity
    - But they may not be compatible

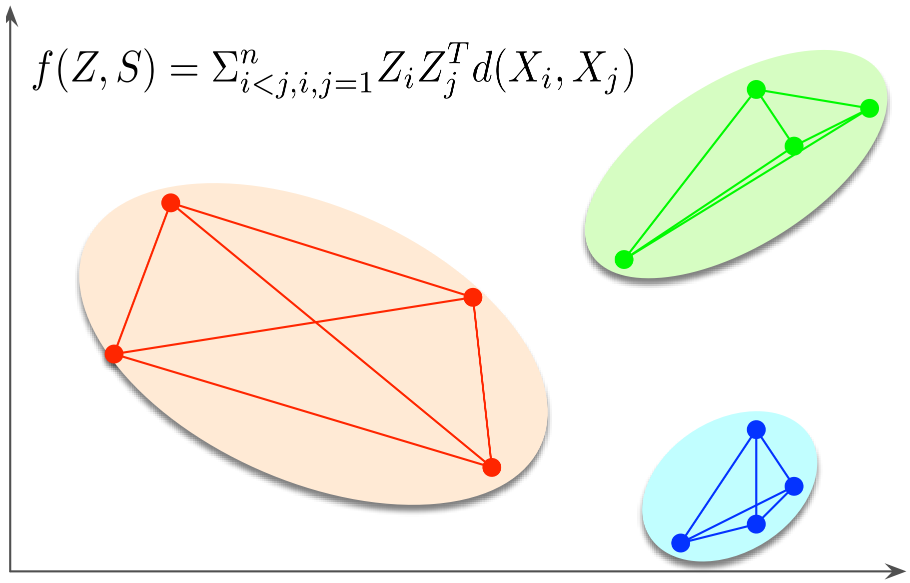
Computing the Pareto Front

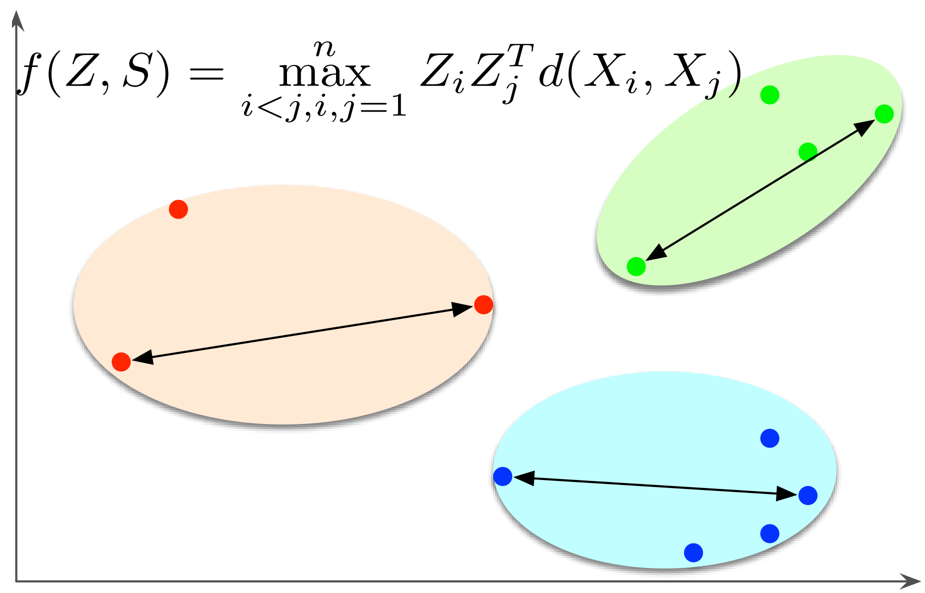
- **If** we knew the Pareto front was convex could just add two objectives with varying weights

Input: Features X, tags D and number of clusters k.
Output: A complete Pareto front $\mathcal{P}$.
$\mathcal{P} \leftarrow \emptyset$;
$s_1^f \leftarrow$ minimize f subject to $\mathcal{C}$;
$i \leftarrow 1$;
while $s_i^f \neq \mathit{NULL}$ **do**
  $s_i^g \leftarrow$ maximize g subject to $\mathcal{C} \cup \{f \leq f(s_i^f)\}$;
  $\mathcal{P} \leftarrow \mathcal{P} \cup \{s_i^g\}$;
  $i \leftarrow i + 1$;
  $s_i^f \leftarrow$ minimize f subject to $\mathcal{C} \cup \{g > g(s_{i-1}^g)\}$;
return $\mathcal{P}$;



Objective f: Cluster Compactness [Dao, Duong, Vrain, AIJ 2017]



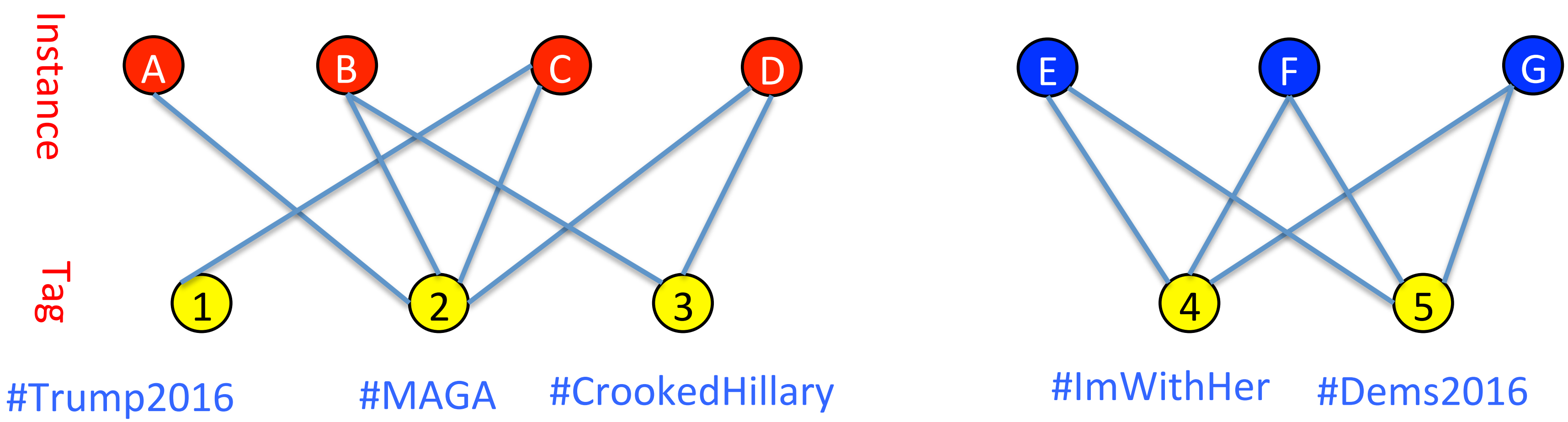


Objective g: Descriptiveness

- Max-Min **Complete** Tag Agreement (MMCTA)
- Minimize Tag α+βDisagreement (α+βMTD):
- Max-Min Neighborhood Agreement (MMNA):

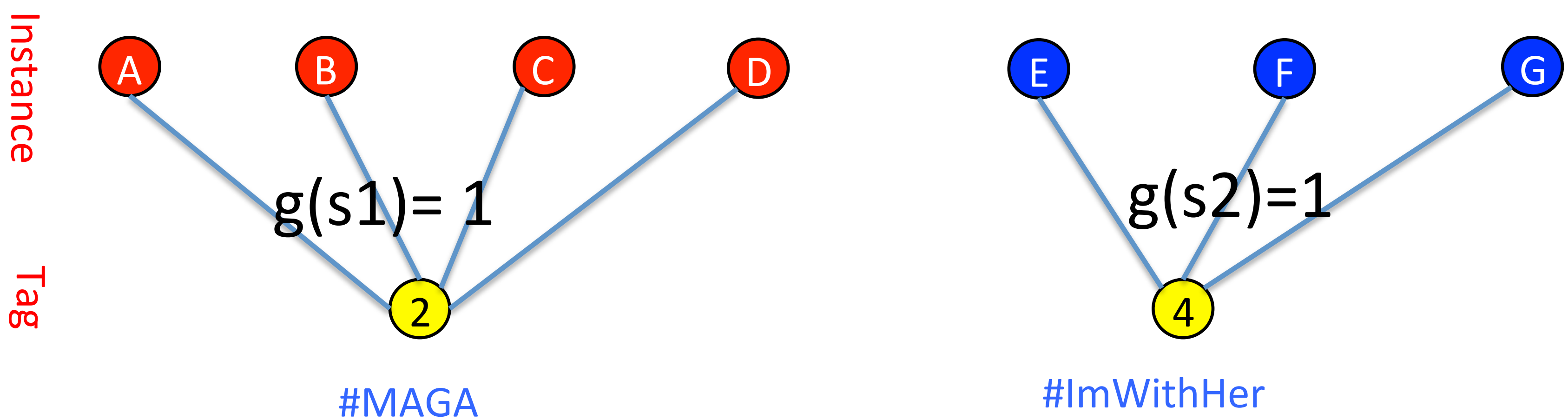
Tightness of Explanation

Use these to choose amongst tags to form a description for each cluster



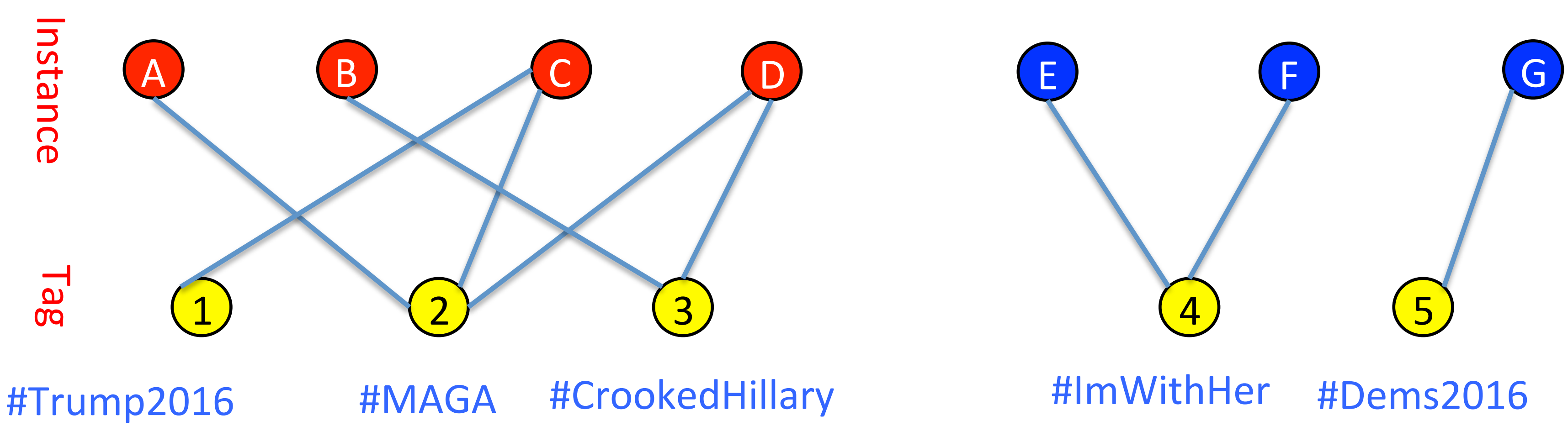
Objective g: Description

- **Max-Min Complete Tag Agreement (MMCTA) - Useful for strong explanations**
- Minimize Tag α+βDisagreement (α+βMTD):
- Max-Min Neighborhood Agreement (MMNA):



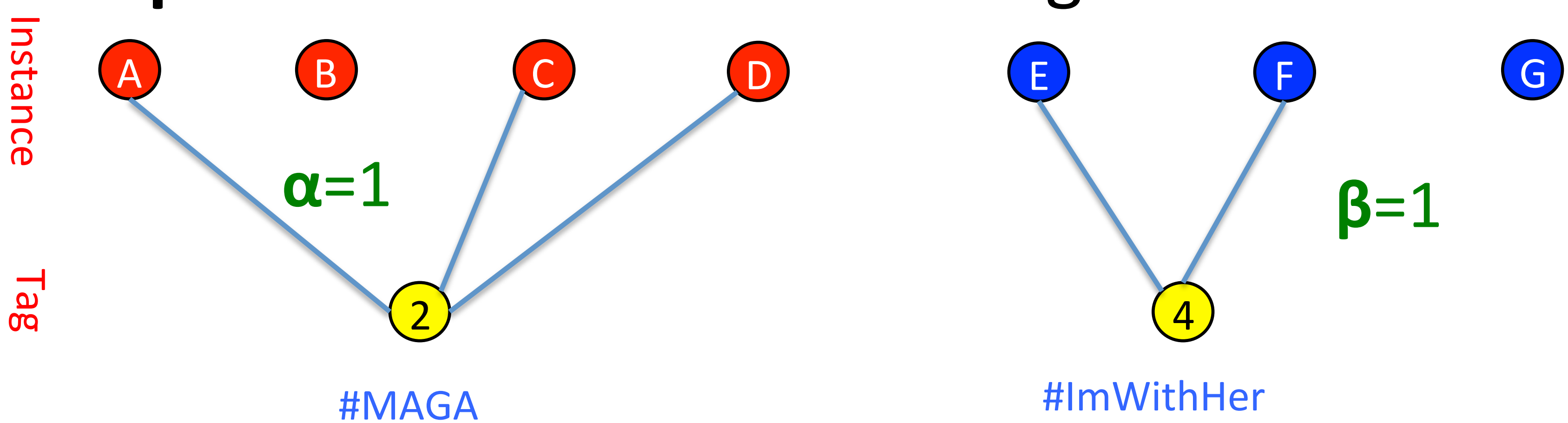
Objective g: Description

- Max-Min Complete Tag Agreement (MMCTA)
- **Minimize Tag α+βDisagreement (α+βMTD):**
- Max-Min Neighborhood Agreement (MMNA):



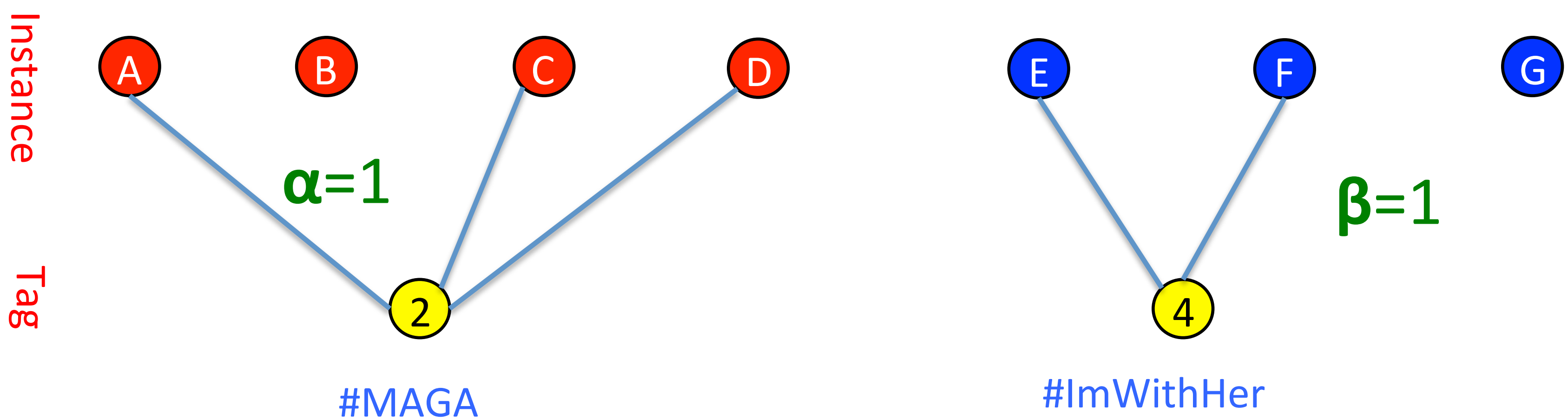
Objective g: Description

- Max-Min Complete Tag Agreement (MMCTA)
- **Minimize Tag α+βDisagreement (α+βMTD):**
- Max-Min Neighborhood Agreement (MMNA):
- **α**= number of tags an instance need not posses
- **β**= number of instances a tag need not cover.



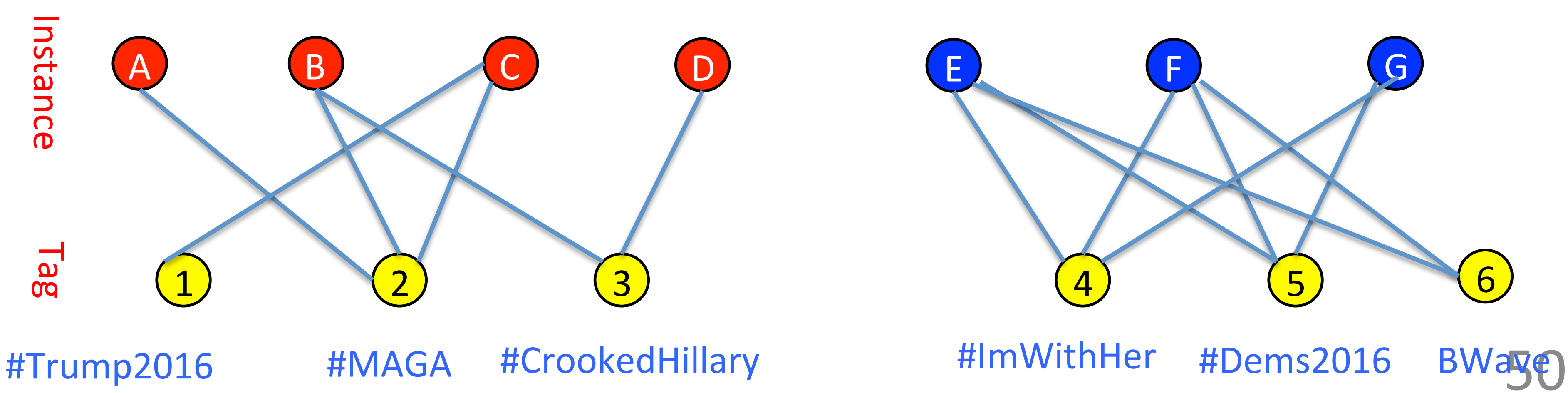
Objective g: Description

- Max-Min Complete Tag Agreement (MMCTA)
- **Minimize Tag α+β Disagreement (α+βMTD):**
- Max-Min Neighborhood Agreement (MMNA):
- **α** for noisy tags i.e. someone doesn't use right tag
- **β** for sparse tags i.e. a tag is useful but rarely used



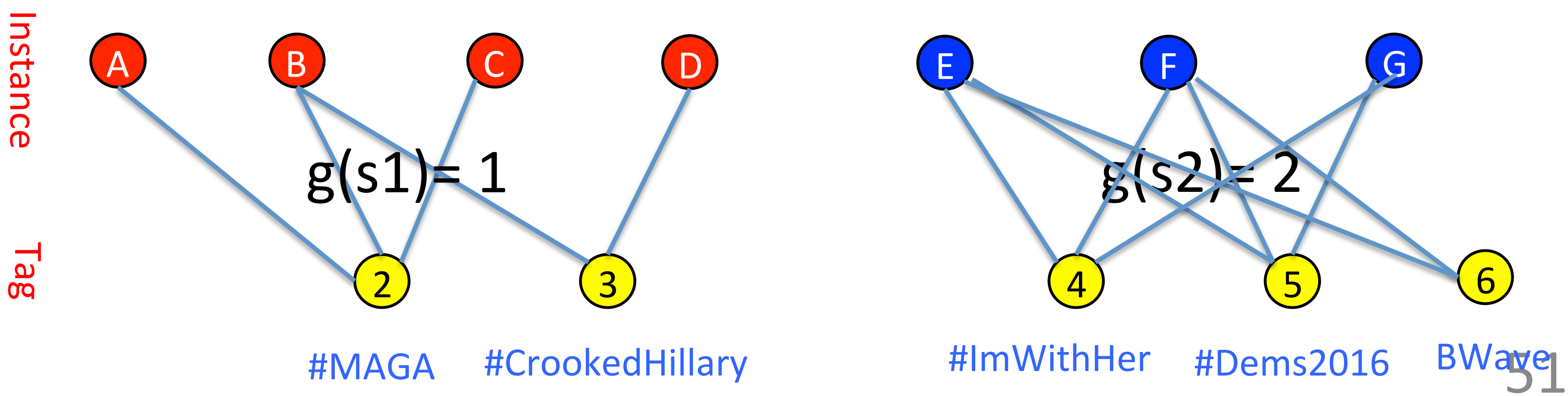
Objective g: Description

- Max-Min Complete Tag Agreement (MMCTA)
- Minimize Tag α+βDisagreement (α+βMTD):
- **Max-Min Neighborhood Agreement (MMNA):**



Objective g: Description

- Max-Min Complete Tag Agreement (MMCTA)
- Minimize Tag α+βDisagreement (α+βMTD):
- **Max-Min Neighborhood Agreement (MMNA):**



Objective g: Description

- Max-Min Complete Tag Agreement (MMCTA)
- Minimize Tag α+βDisagreement (α+βMTD):
- Max-Min Neighborhood Agreement (MMNA):

Theorem 1 *The MMCTA problem is* **NP**-*complete even when* $q = 1$.

Theorem 2 *(a) For any* $k \geq 3$*, the MMNA problem is* **NP**-*complete even when* $q = 1$. *(b) The problem is efficiently solvable for* $k = 2$.

Multi-Objective Formulation

Each image described by SIFT features and variable number of tags (i.e. fuzzy, tail, etc.). **Animal names not given to algorithm**

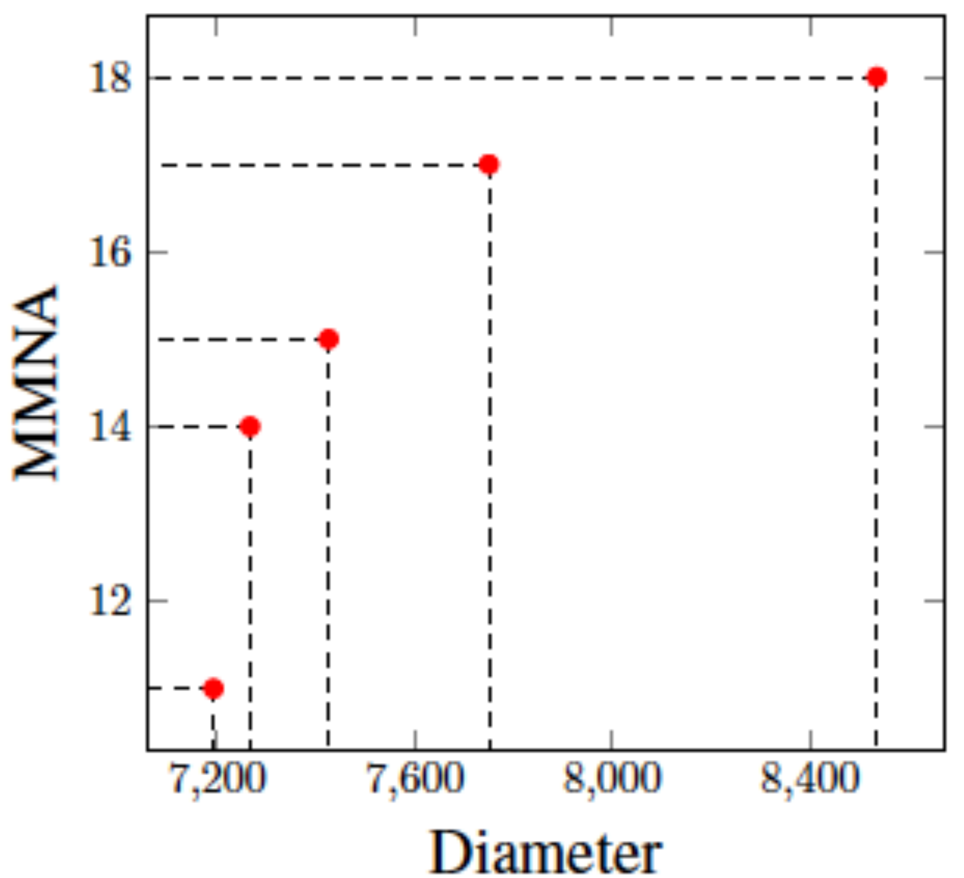
[Lampert, C. H.; Nickisch, H.; and Harmeling, S. CVPR 2009]

Trade off Compactness vs Useful Description

Cl#	Composition by animals	Description by tags
C1	1 grizzly bear, 2 dalmatian, 1 horse, 2 blue whale	...e, newworld, smart
C2	5 antelope, 2 grizzly bear, ... cat	... quadrapedal, new-
C3	69 beaver, 64 dalmatian, 4...	...nid, smart, solitary
C4	100 killer whale, 69 blue w...	
C5	95 antelope, 97 grizzly be... siamese cat	... quadrapedal, new-

Cl#	Composition by animals	Description by tags
C1	2 antelope, 4 dalmatian, 2 ... man shepherd, 4 siamese c...	...ewworld, oldworld,
C2	2 beaver, 1 persian cat, 1 ... man shepherd	
C3	100 grizzly bear, 98 beave... cat, 1 siamese cat	...y
C4	100 killer whale, 100 blue ...	...ish, plankton, arctic,
C5	98 antelope, 96 dalmatian, ... german shepherd, 95 siam...	...ewworld, oldworld,

Cl#	Composition by animals	Description by tags
C1	100 antelope, 100 dalmati...	...e, agility, newworld,
C2	100 horse, 99 german s... siamese cat	...quadrapedal, active,
C3	100 grizzly bear, 100 beav... cat	..., newworld, ground,
C4	100 killer whale, 100 blue ...	...ish, plankton, arctic,
C5	100 persian cat, 1 german shepherd, 1 siamese cat	gray, furry, pads, paws, tail, chewteeth, meatteeth, claws, walks, fast, quadrapedal, agility, meat, newworld, oldworld, ground, smart, solitary, domestic



(c) Fifth Pareto point: MMNA maximized. MMCTA=15, MMNA=18

Limitations

- Pareto optimization using the method is really slow
  - Efficient ways to compute a subset of the Pareto front?
- How to use or interpret the Pareto front?
  - Counter to most ML which gives you a single result, here we have a range of results

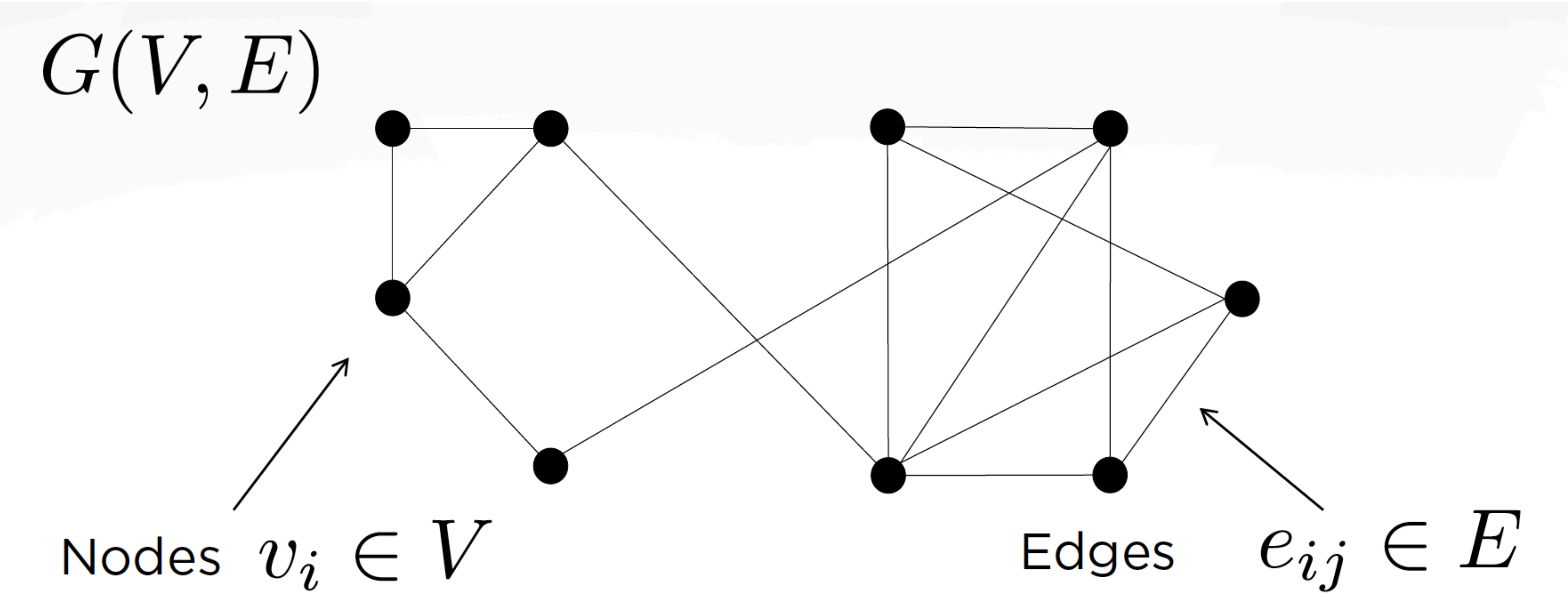
Plan For The Clustering Module

- Adding **explanation** to clustering
  - How can we make clustering explainable
- **Classic formulations of clustering**
  - **Correlation clustering**
- Using CP to add **constraints** to clustering
- **Human in the loop** style minimal modification clustering

Correlation Clustering

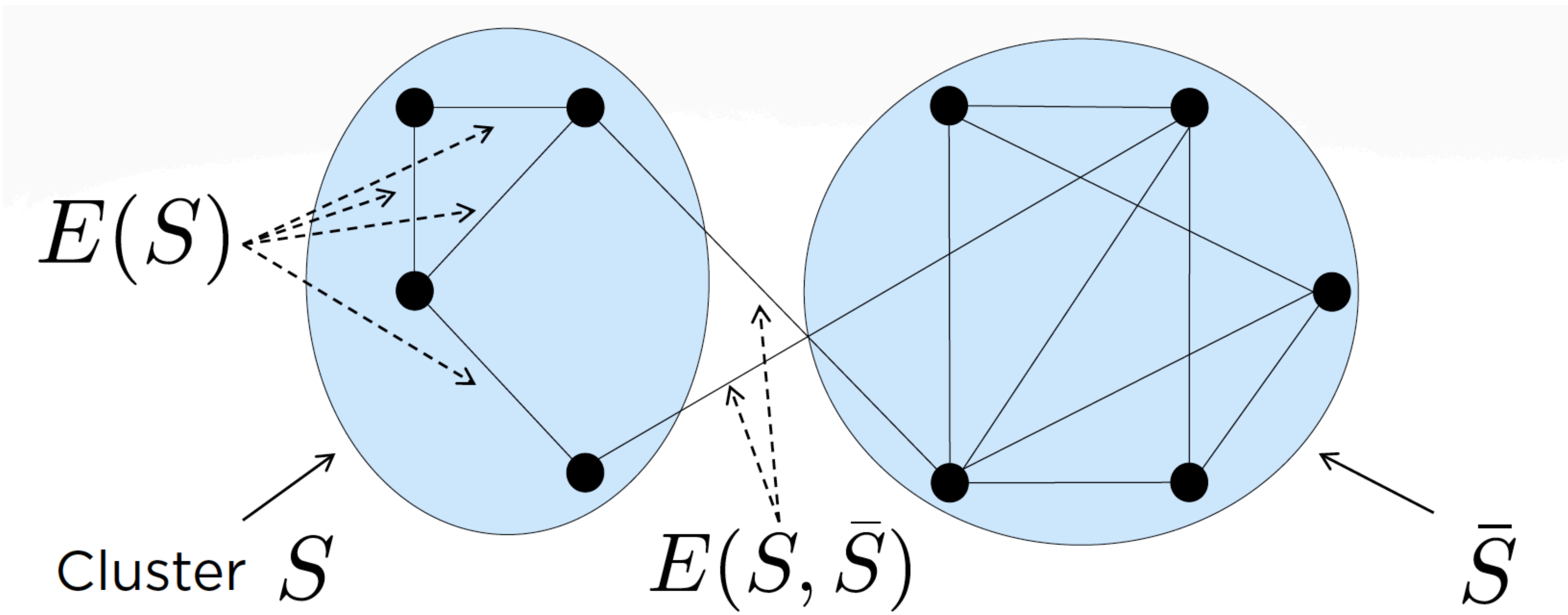
✔ Classic Setting	Extending Classic Settings	Completely New Settings
✔ Pure SAT/MIP/ CP Formulation	Extensions To SAT/MIP/CP Solvers	Hybrid Formulations

Correlation Clustering Versus Graph Partitioning



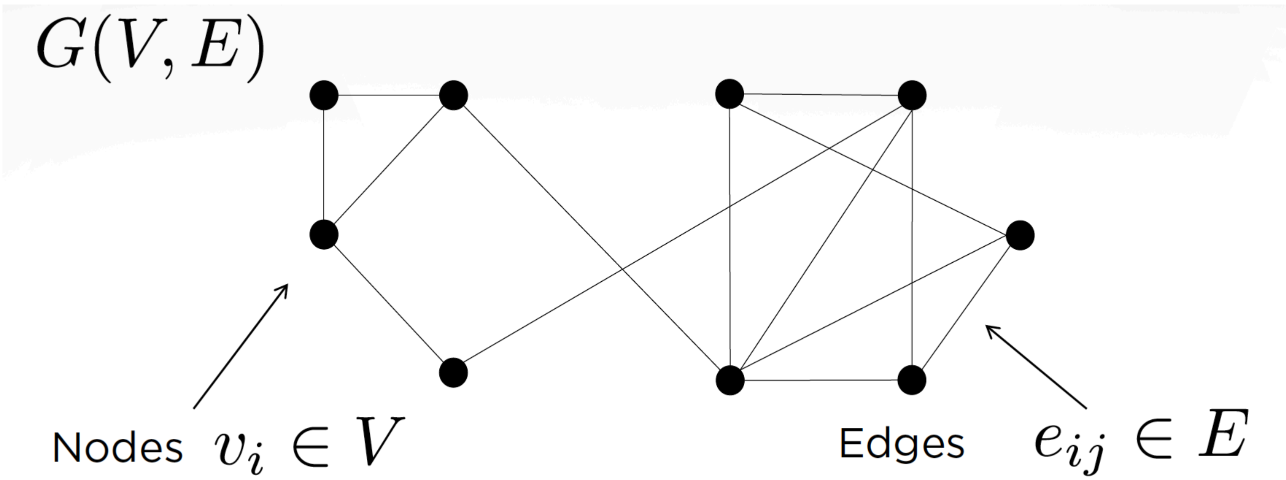
$e_{ij} \in E$ means the two nodes are “similar”

An Objective

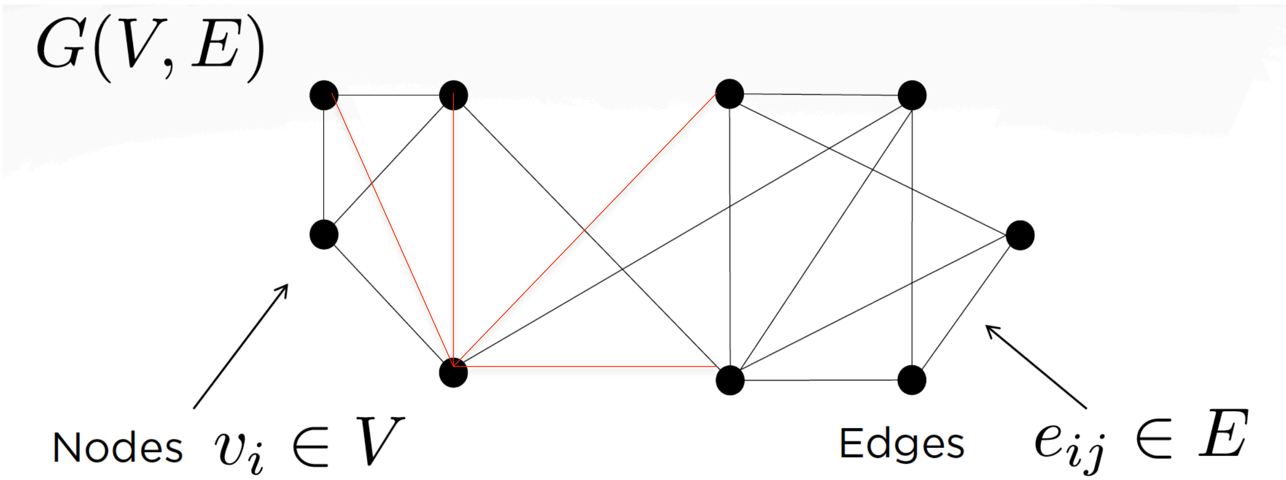


Sparsest cut objective $$\frac{|E(S, \bar{S})|}{|S| \cdot |\bar{S}|}$$

Correlation Clustering Also Considers Negative Edges

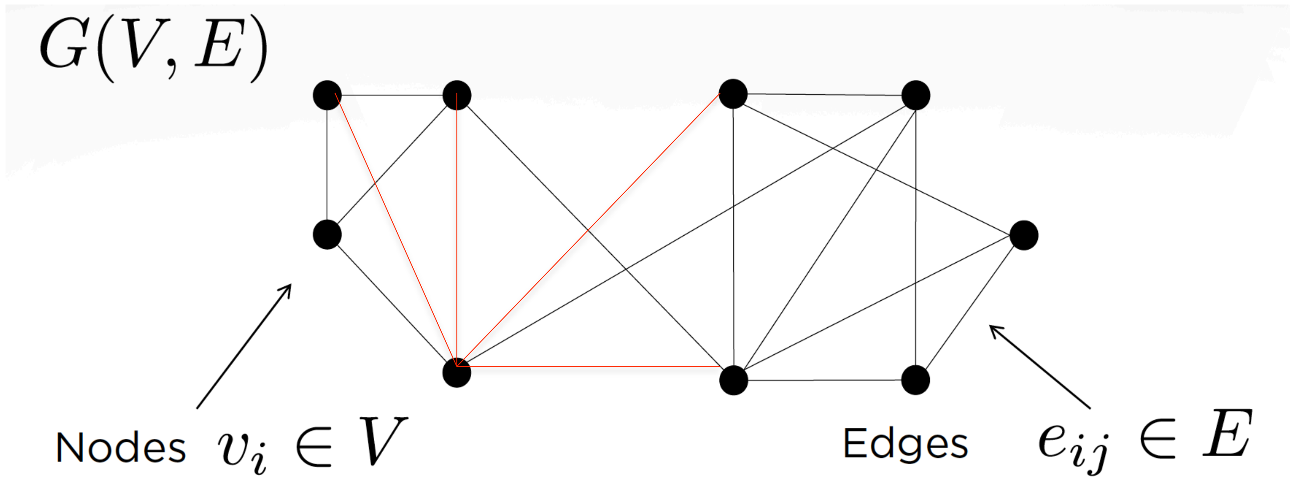


Correlation Clustering Also Considers Negative Edges

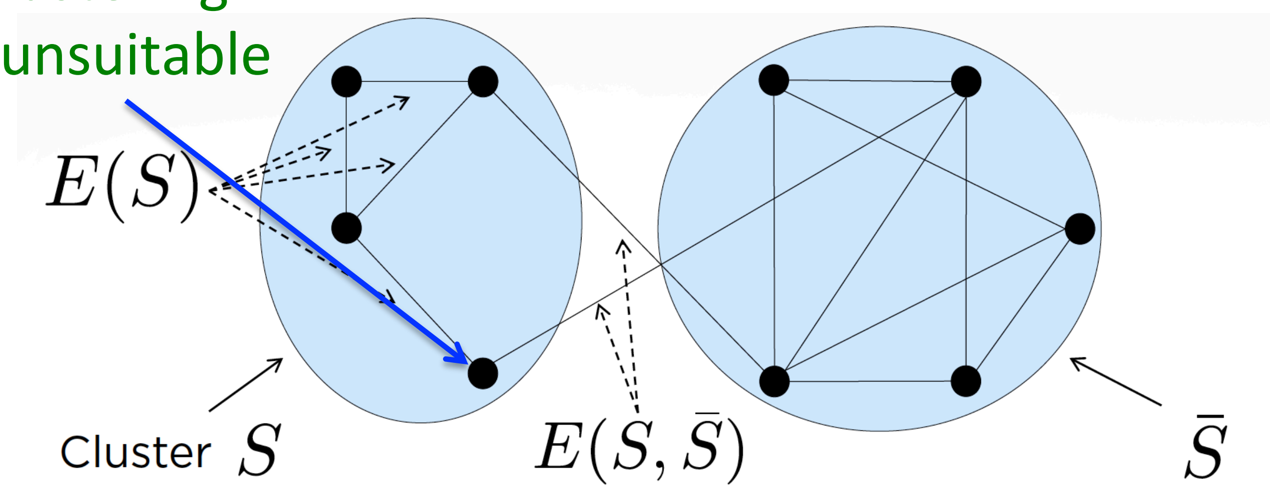


Cut solution?

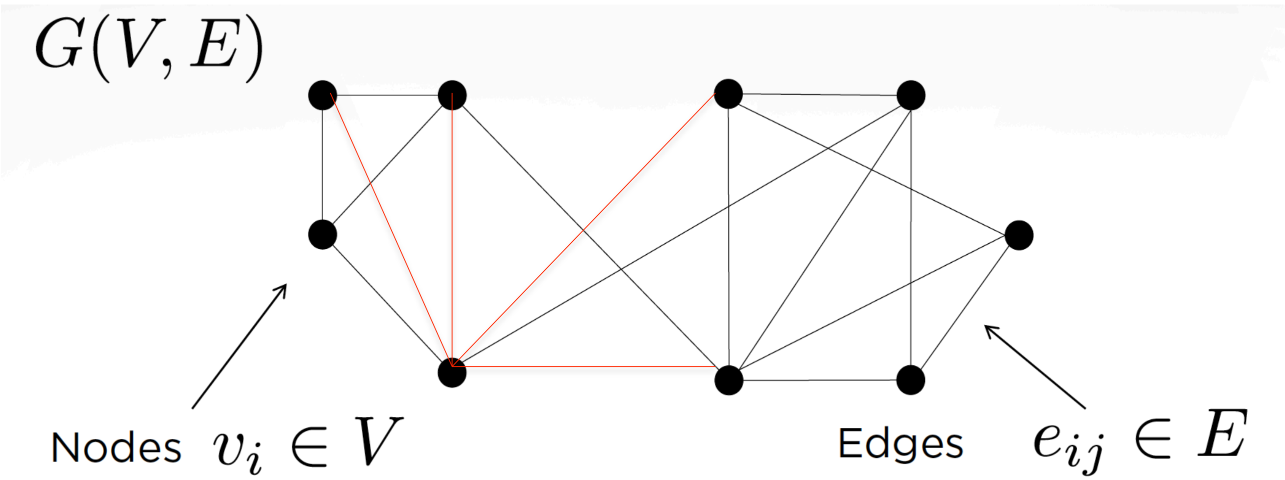
Correlation Clustering Also Considers Negative Edges

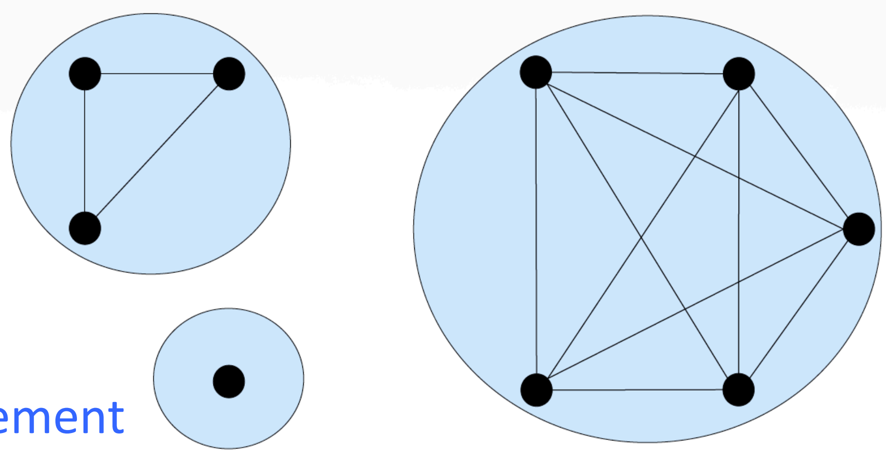


This clustering
Seems unsuitable



Correlation Clustering Also Considers Negative Edges





This seems more reasonable

Effectively Maximize/Minimize Agreement/Disagreement b/w **Given Graph** and **Induced Graph**

Formalizing Problem

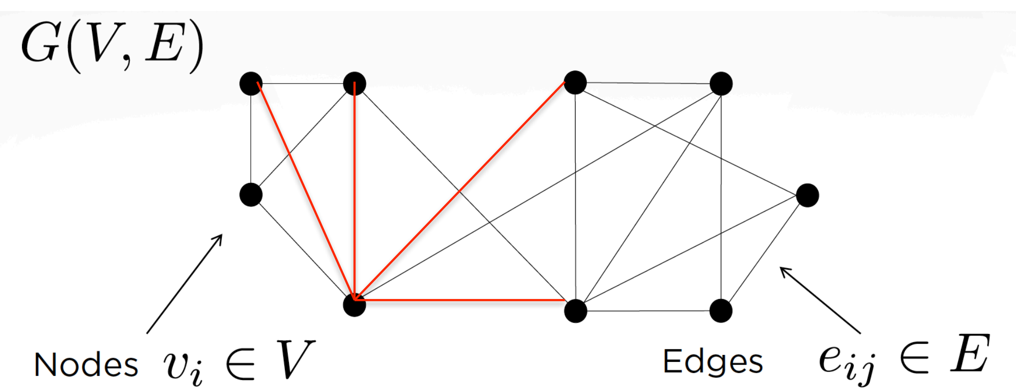
	Unweighted
Min-disagree	$$\min_C \sum_{i,j} C_{ij}(1 - E_{ij}) + \sum_{i,j}(1 - C_{ij})E_{ij}$$
Max-agree	$$\max_C \sum_{i,j} C_{ij}E_{ij} + \sum_{i,j}(1 - C_{ij})(1 - E_{ij})$$

"Adjacency" Matrix Given Graph

E_{ij} = +1/0 for a +ve/-ve edge

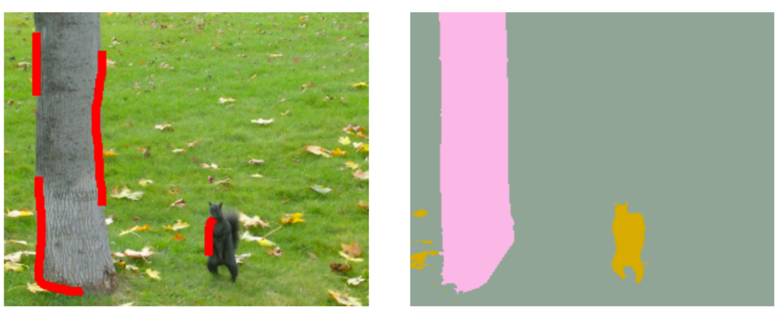
"Adjacency" Matrix Induced Graph

C_{ij} = 1/0 for two points in the same/different cluster



Lots of Applications

YAHOO!



(a) Input image and boundary scribbles (red)

(b) Resulting segmentation

Figure 3: **Interactive multi-object segmentation:**

Results So Far From Theoretical CS

- Correlation clustering is NP-hard [BBC02]
- Constant factor approximation based on simple rounding scheme [BBC02]
- Simple randomized algorithm produces a 3 approximation with linear run-time [ACN05]
- Best approximation is a 2.06 factor use LP then rounding
- The next formulation will solve this problem exactly

MIP Formulation

[Berg, Jeremias, and Matti Jarvisalo. "Optimal correlation clustering via MaxSAT, AIJ Journal 2017](#)

- Let x_{ij}=1 if instance *i* and *j* put in same cluster
- Let s() =1 if instance *i* and *j* have positive edge
- Let s() =0 if instance *i* and *j* have negative edge

Dis-agreements

Agreements

$$\text{MINIMIZE} \quad \sum_{s(v_i,v_j)=0} x_{ij} - \sum_{s(v_i,v_j)=1} x_{ij}$$

$$\text{where} \quad x_{ij} + x_{jk} \leq 1 + x_{ik} \text{ for all distinct } i, j, k$$

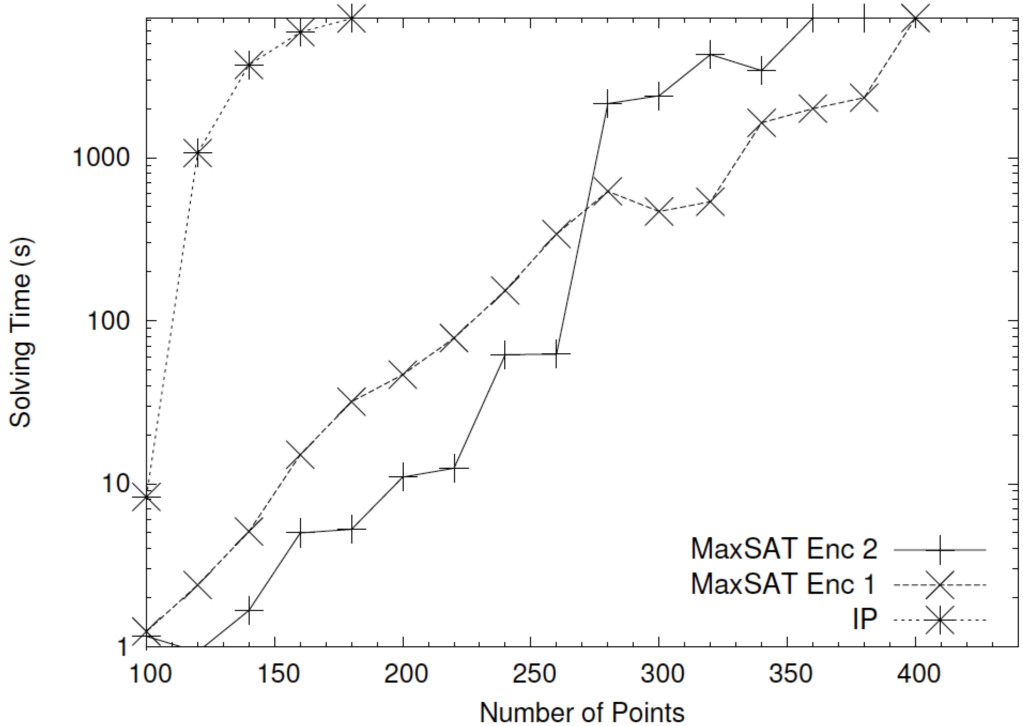
$$x_{ij} \in \{0,1\} \text{ for all } i,\ j \text{ s.t } i \neq j.$$

"attempts to optimally solve this integer program have not so far been reported"

MAXSAT Formulation

[Berg, Jeremias, and Matti Jarvisalo. "Optimal correlation clustering via MaxSAT, AIJ Journal 2017]()

Hard Clauses F_h^1:	$(\neg x_{ij} \vee \neg x_{jk} \vee x_{ik})$	for all $(v_i, v_j, v_k) \in V^3$ where i, j, k are distinct
Soft Clauses F_s^1:	(x_{ij})	for all $s(v_i, v_j) = 1$
	$(\neg x_{ij})$	for all $s(v_i, v_j) = 0$



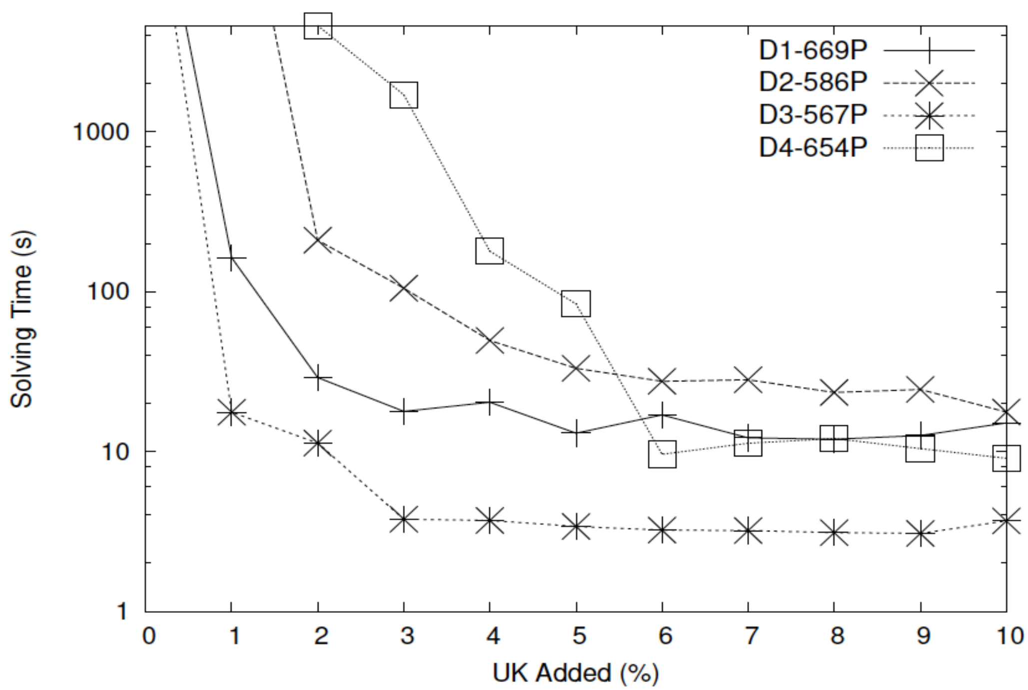
+ Does not require k to be specified

- Number of transitivity constraints

Benefits of Adding Guidance

Berg, Jeremias, and Matti Jarvisalo. "Optimal correlation clustering via MaxSAT, AIJ Journal 2017

- Benefit of MAXSAT formulation: adding composition constraints: x_{ij}= T, x_{jk}= F
- Results in runtime going substantially down



Open Questions

Can we derive partial solutions to "seed" the solver?

Can we formalize IF adding constraints will produce the most speed up? i.e. graph properties

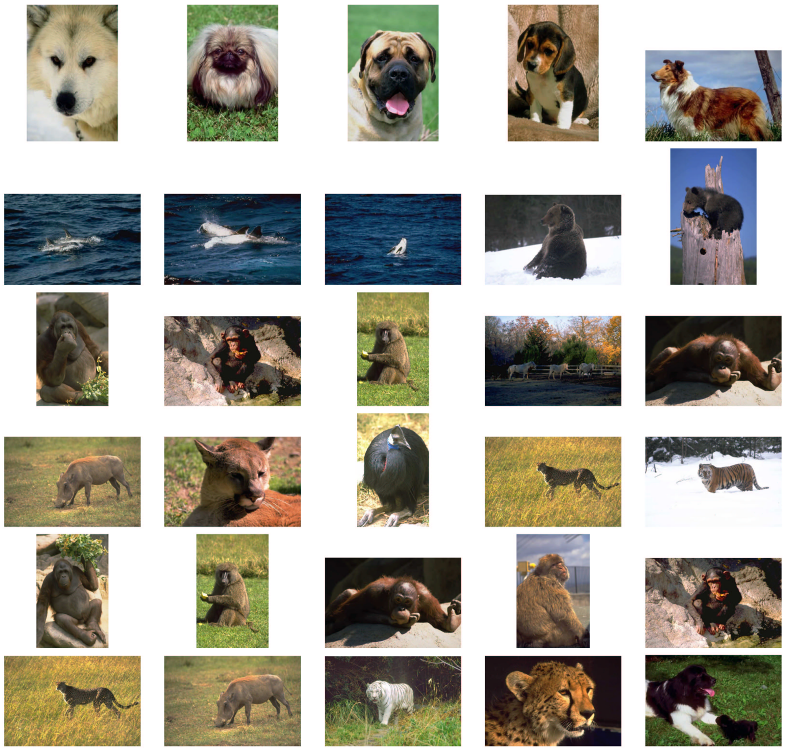
Can we formalize WHICH constraints will produce the greatest speedup. i.e. active setting

Plan For The Clustering Module

- Adding **explanation** to clustering
  - How can we make clustering explainable
- **Classic** formulations of clustering
  - Correlation clustering
- **Using CP to add constraints to clustering**
- **Human in the loop** style minimal modification clustering

Adding Constraints to Clustering

Consider These Problems



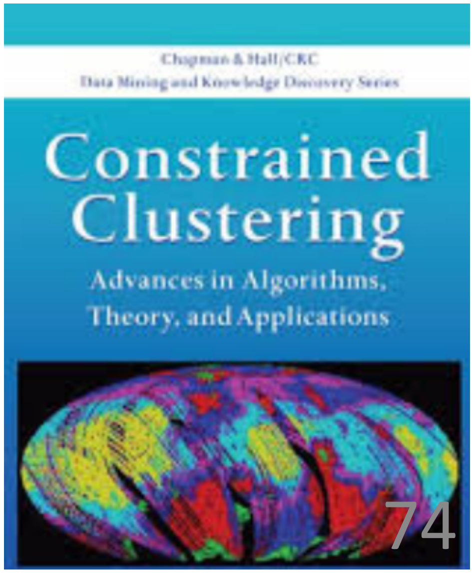
Consider These Problems

Now With Guidance/Constraints/Side-Info

Maximum width of a lane is 3 meters

Brief History

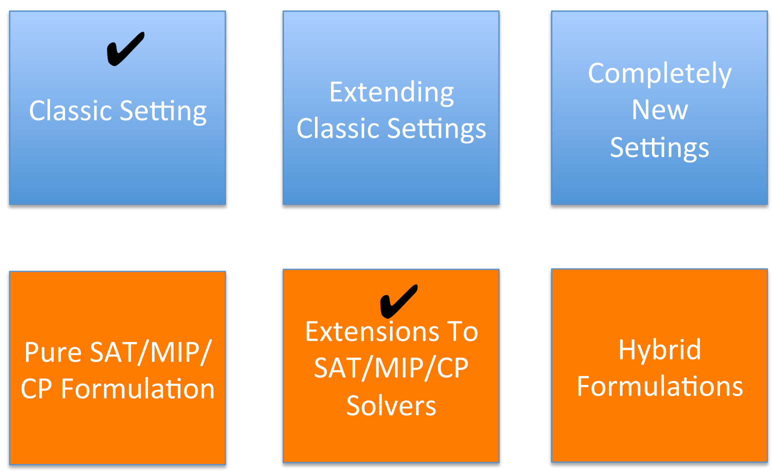
- 2000's lots of work on adding side information (modeled as constraints) to learning
  - Recent attention by parts of CP community.
- Why? Regular Clustering is fine in **knowledge poor** domains such as Google (ads), Amazon/Netflix (recommenders).
  - No need to be consistent with or satisfy domain side information.
  - But in domain/knowledge rich areas ...
- Benefits? Compensates for
  - Improves performance c.f. ground truth
  - Lack of labeled data with weaker guidance
  - Objective function being limited
- I'm talking on Tuesday at DL + CC Tuesday



Limitations of Existing Work

- Simple conjunctions of simple constraints
  - Big AND of together/apart pairwise constraints
  - Why? Mechanism to encode constraints is typically a graph/matrix
  - Difficult to encode anything but: “satisfy all constraints”
  - Other complex constraints could we place on clusters
    - Cardinality, Density etc.
- Relaxation of problem so not finding global optima
  - i.e. Spectral clustering only finds min-cut for two-way cut
- Optimizes simple objective functions
  - Hope the objective function matches real world criteria

CP Formulations for Constrained Clustering

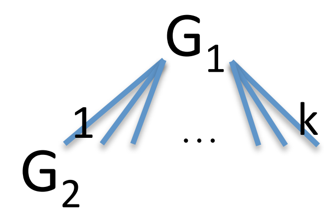


CP Formulations

Bich Duong, Khanh-Chuong, and Christel Vrain. "A declarative framework for constrained clustering". ECML, 2013. … AIJ 2017 [http://cp4clustering.com/]

- Decision variables $G_1 \dots G_n$
  - Dom(G_1) = { 1...k}
  - No need to enforce transitivity.

Search Tree



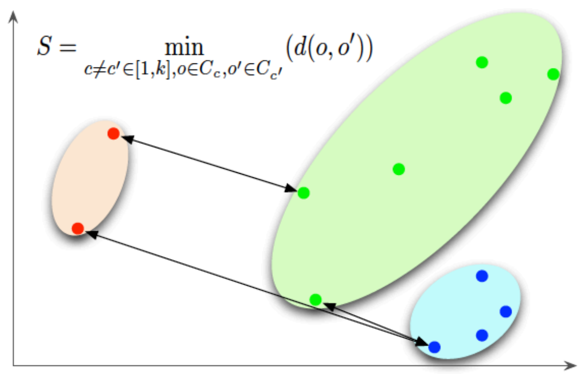
- But lots of symmetry
  - First point always in cluster 1
  - Cluster k is never used until predecessors are used

$$\forall i \in [2, N], \text{ if } G_i = k \text{ then } \exists j < i \text{ st. } G_j = k - 1$$

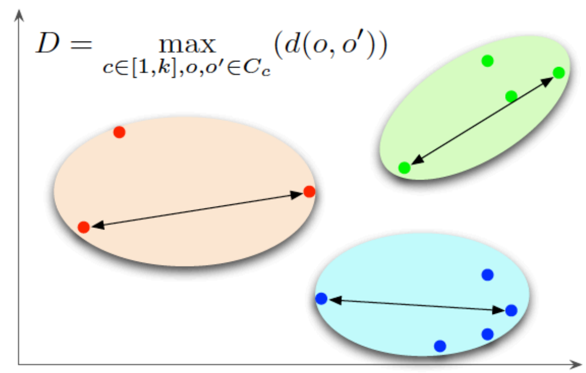
$$Precede([G_1, .., G_N], [1, K_{max}])$$

Optimization Criteria

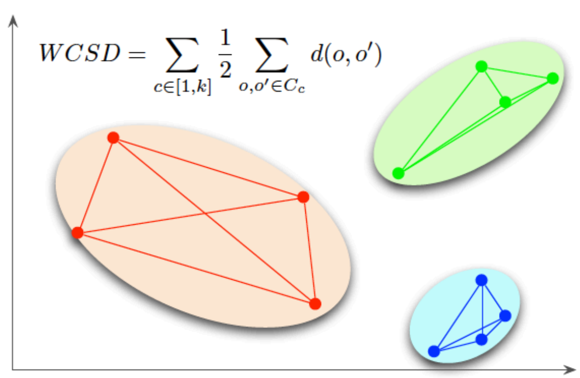
[Bich Duong, Khanh-Chuong, and Christel Vrain. "A declarative framework for constrained clustering”. ECML, 2013. and AIJ 2017]



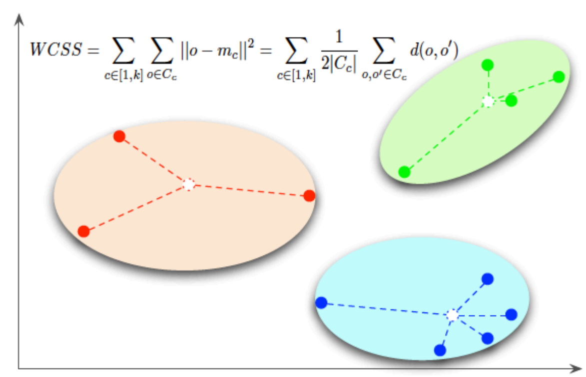
Maximizing the minimal split



Minimizing the maximal diameter



Minimizing the WCSD



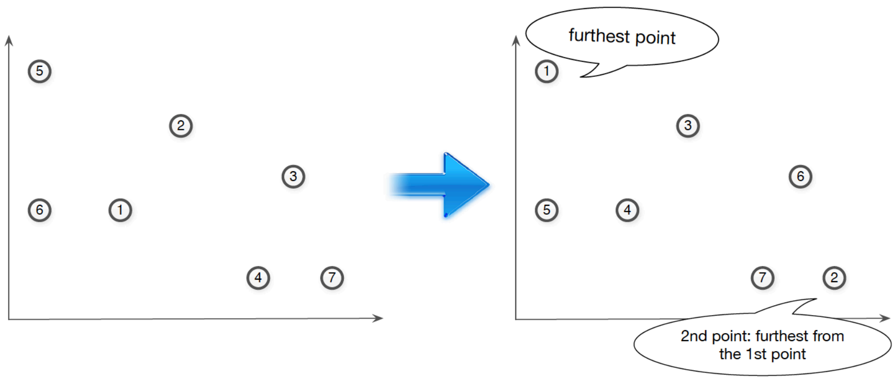
Minimizing the WCSS

Search Strategies

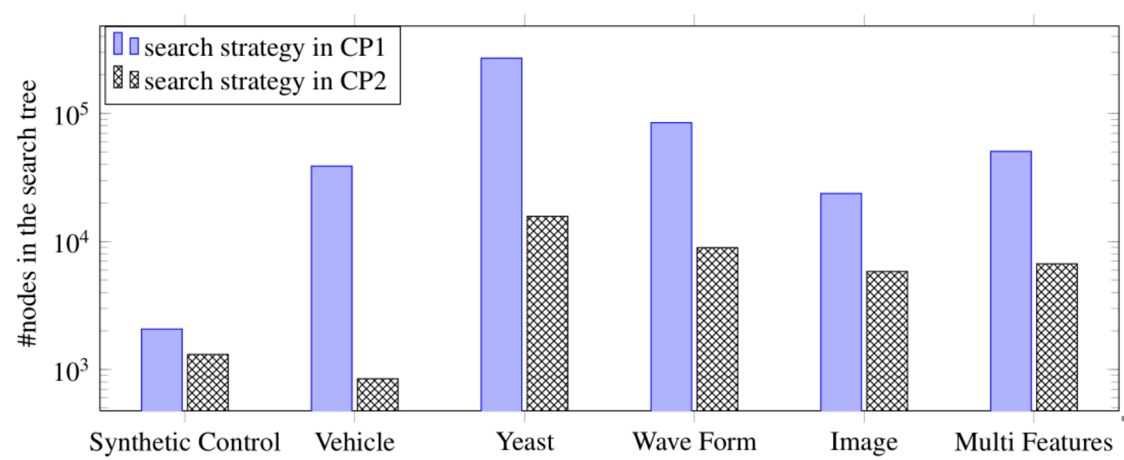
[Bich Duong, Khanh-Chuong, and Christel Vrain. "A declarative framework for constrained clustering". ECML, 2013. and AIJ 2017]

Tree search: instantiating G's in random order?

– Use further point first ordering



Points are then ordered and indexed, so that points that are probably representatives have small index



Pruning the Domains of Variables

[Bich Duong, Khanh-Chuong, and Christel Vrain. "A declarative framework for constrained clustering". ECML, 2013. and AIJ 2017]

- Bounds on min-max cluster diameter D_{Opt}
- Let the FPF of first k points be D_{FPF}
  - Then $D_{Opt} < D_{FPF} < 2D_{Opt}$ [Gonzalez 1985]
  - Then D_{Opt} \in $[D_{FPF} / 2 , D_{FPF}]$
  - Hence \forall
    - Lower Bound Test: $D(i,j) < D_{FPF} / 2 \Rightarrow G_i = G_j$
    - Upper Bound Test: $D(i,j) > D_{FPF} \Rightarrow G_i <> G_j$

Good News and Bad News

Bich Duong, Khanh-Chuong, and Christel Vrain. "A declarative framework for constrained clustering". ECML, 2013. and AIJ 2017

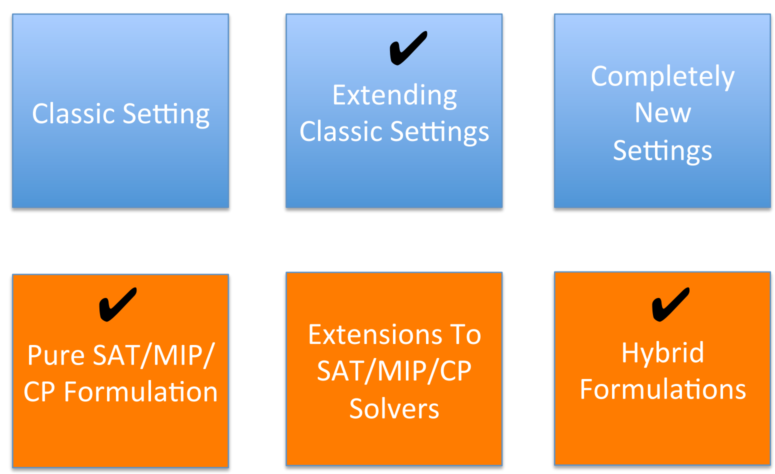
BaB: Branch-and-Bound approach (Brusco et al., 2003)
GC: Algorithm based on graph coloring (Hansen et al., 1978)
CP1: direct modeling (DDV, 2013)
CP2: modeling using dedicated global constraint (DDV, 2015)

Dataset	D_{opt}	BaB	GC	CP1	CP2
Iris	2.58	1.4	1.8	< 0.1	< 0.1
Wine	458.13	2	2.3	< 0.1	< 0.1
Glass	4.97	8.1	42	0.4	0.2
IonoSphere	8.6	–	0.6	0.3	0.3
User Knowledge	1.17	–	3.7	15.4	0.2
Breast Cancer	2377.96	–	1.8	0.7	0.4
Synthetic Control	109.36	–	–	23.6	1.6
Vehicle	264.83	–	–	11.9	0.9
Yeast	0.67	–	–	574.2	5.2
Multi Features	12505.5	–	–	–	10.4
Image Segmentation	436.4	–	–	226.7	5.7
Waveform	15.6	–	–	–	50.1

Some Directions

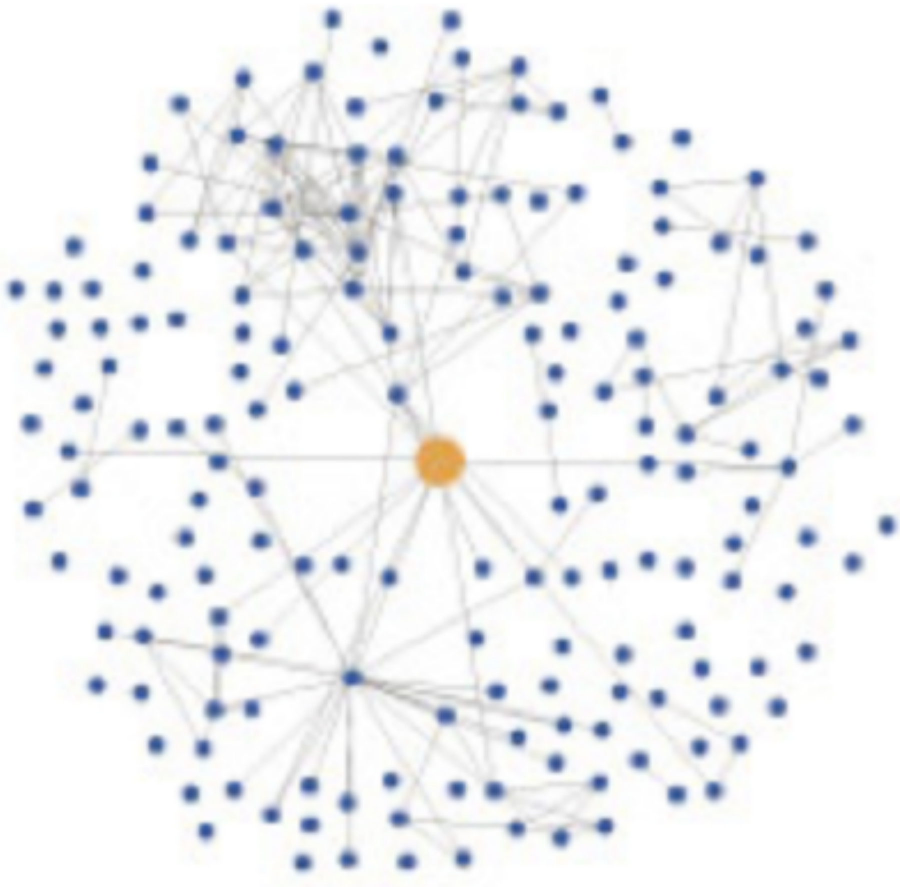
- No need to believe human constraints are compatible with optimizing objective
  - Treat as Pareto optimization problem?
- Adding in more complex constraints
  - Complex combinations of instance level constraints
  - Cluster level constraints
  - Constraints b/w clusters

New Types of Constraints



Imagine This Dinner Parties Problem

- Your new years resolution − Be more social?

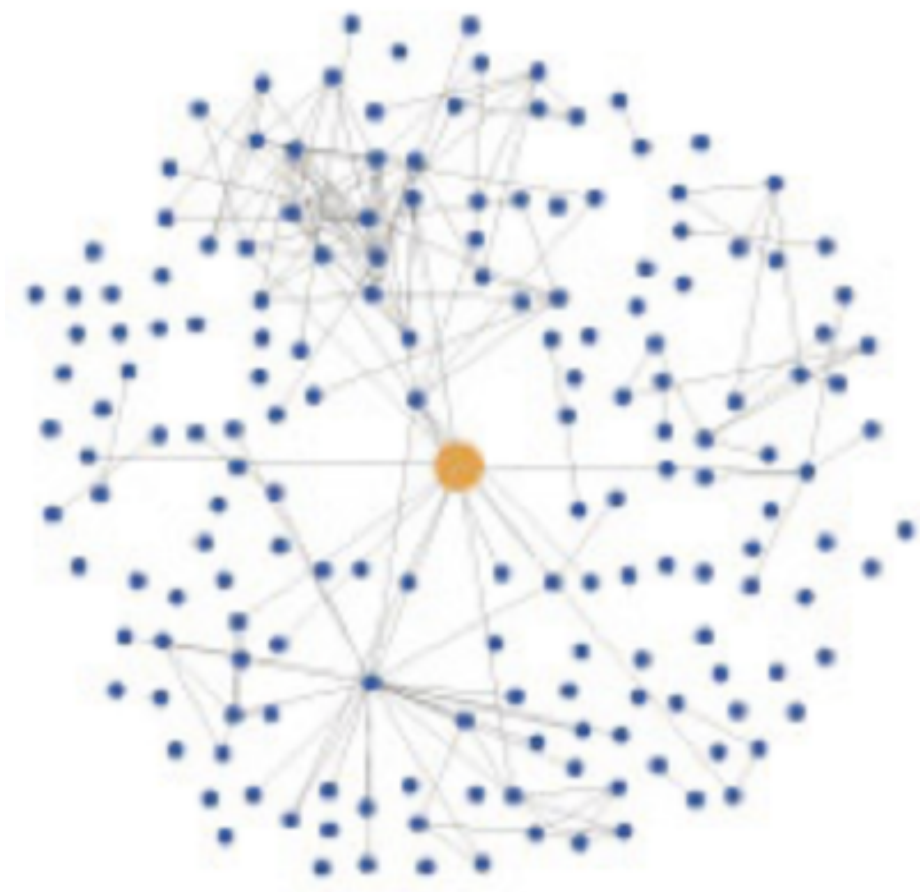


Partition to get friends-of-friends?

Create k parties, but **avoid parties of**
i) Mostly men/woman
ii) Wide age discrepancy
iii) Of people with nothing in common

Imagine This Dinner Parties Problem

- Your new years resolution – be more social



Partition to get friends-of-friends?

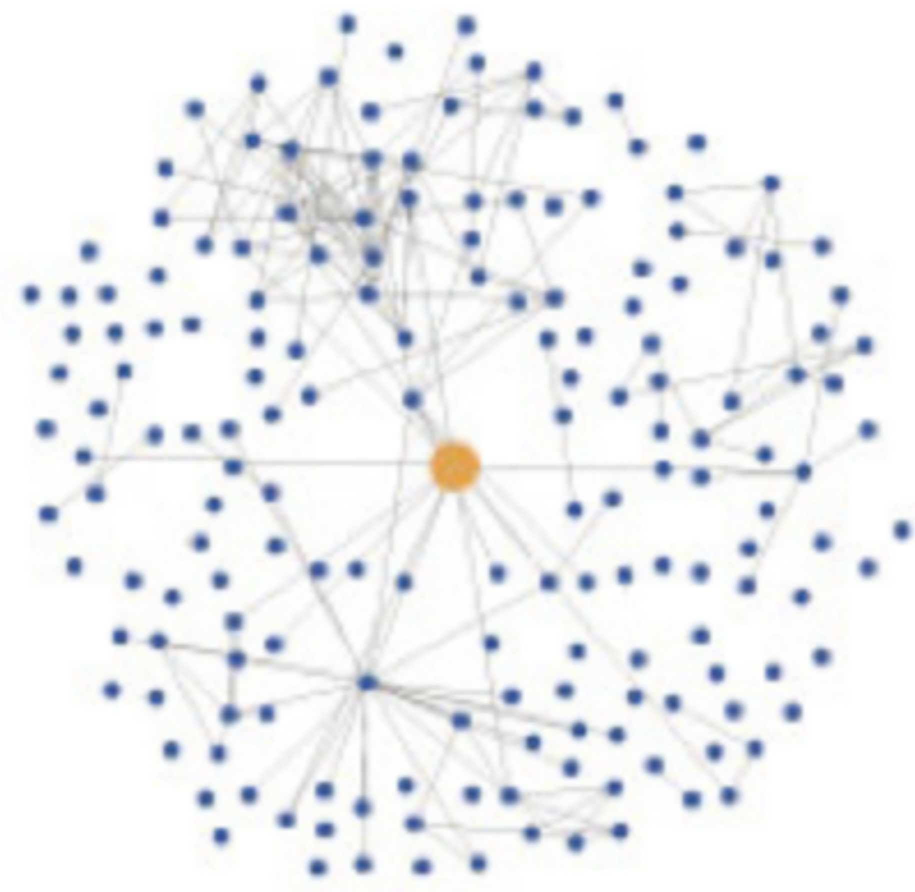
Like to require for each party:

a) Balance males/females
b) Diameters on properties like age
c) Density reqs. w.r.t. to interests

Each of these could be a constraint or another objective

Beyond Pairwise Constraints

[Duong et. al. "A Framework for Actionable Clustering". ECAI 2016]



Partition to get friends-of-friends?

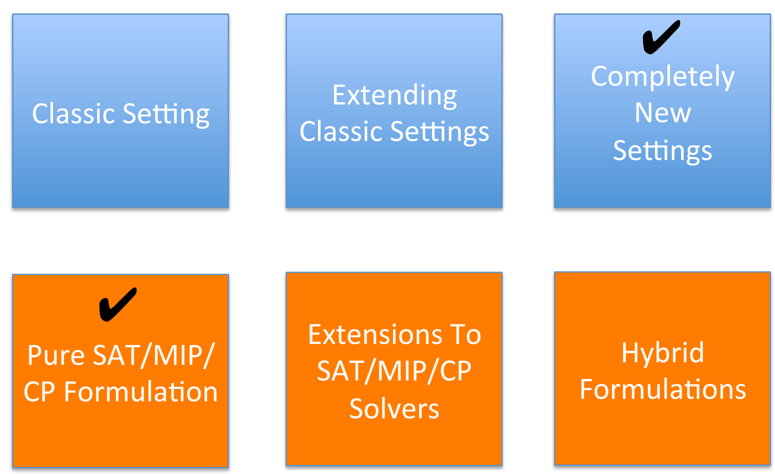
New Types of Constraints

a) Cardinality: Balance males/females
b) Geometric: Diameters
c) Density: Minimum relations

Plan For The Clustering Module

- Adding **explanation** to clustering
  - How can we make clustering explainable
- **Classic** formulations of clustering
  - Correlation clustering
- Using CP to add **constraints** to clustering
- **Human in the loop style minimal modification clustering**

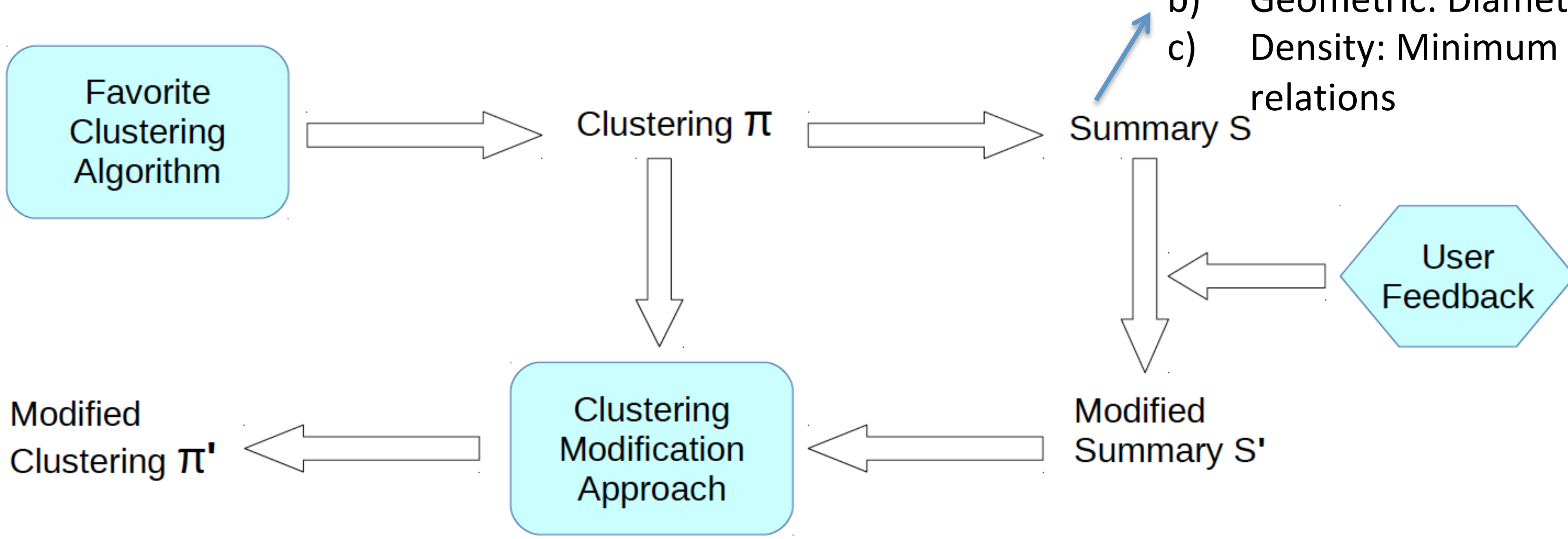
Minimal Modification Problem



Problem Setting

[A Framework for Minimal Clustering Modification via Constraint Programming, Tom Kuo et. al. AAAI 17](#)

a) Cardinality: Balance males/females
b) Geometric: Diameters
c) Density: Minimum relations



Minimally modify Π to obtain Π’ to satisfy S’

$$\begin{aligned} \underset{\Pi'}{\text{minimize}} \quad & d(\Pi, \Pi') \\ \text{subject to} \quad & \Pi' \text{ satisfies } S' \end{aligned}$$

Intractable Problem

[A Framework for Minimal Clustering Modification via Constraint Programming, Tom Kuo et. al. AAAI 17]

Theorem (1)

The reclustering problem where $\ell = 2$ is NP-complete.

Proof idea: reduction to Covering Points by Unit Squares.

Even for very limited settings

Theorem (2)

Suppose the number of dimensions along which the maximum diameter must be reduced is a variable ℓ. The reclustering problem is NP-complete for any $k \geq 3$.

Proof idea: similarly reduction to Covering Points by Unit Hypercubes.

Formulation

[A Framework for Minimal Clustering Modification via Constraint Programming, Tom Kuo et. al. AAAI 17]

$$\underset{z,C,L,H,\Pi'}{\texttt{minimize}} \quad \sum_{i=1}^{n} z[i]$$

Number of modifications

$$\texttt{subject to}$$

$$\forall c = 1, \ldots, k, \ \forall i = 1, \ldots, n, \ C[c,i] = \mathbb{I}[\Pi'[i] = c]$$

$$\forall i = 1, \ldots, n, \ z[i] = \mathbb{I}[\Pi'[i] \neq \Pi[i]]$$

$$\forall c = 1, \ldots, k, \ \ \forall t = 1, \ldots, f,$$

$$L[c,t] = \underset{i=1,\ldots,n}{\texttt{min}} \{C[c,i](X[i,t] - M_u[t])\} + M_u[t]$$

$$H[c,t] = \underset{i=1,\ldots,n}{\texttt{max}} \{C[c,i](X[i,t] - M_l[t])\} + M_l[t]$$

Smallest/largest values for t^th feature

$$H[c,t] - L[c,t] \leq \mathcal{D}'[c,t]$$

User given constant

Results

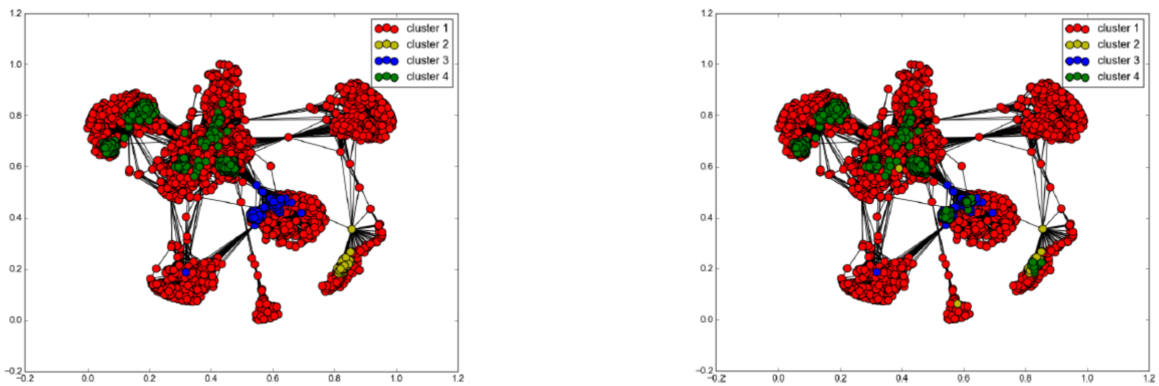
A Framework for Minimal Clustering Modification via Constraint Programming, Tom Kuo et. al. AAAI 17

Data: Facebook egonets[1]

Initial clustering: 4-way clustering from spectral clustering on *friendship graph*

Modification: balance (i.e. bounds diameters) two features/dimensions, gender and some language

Results:



(a) Initial (b) Modified

Figure: Visualization of clusterings on Facebook egonets graph.

References – SAT and Clustering

- Berg, Jeremias, and Matti Jarvisalo. "Optimal correlation clustering via MaxSAT." *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*. IEEE, 2013. 13.pdf

- Berg, Jeremias, Antti Hyttinen, and Matti Järvisalo. "Applications of MaxSAT in data analysis." *Proceedings of the 6th Pragmatics of SAT Workshop*. 2015.

- Davidson, Ian, S. S. Ravi, and Leonid Shamis. "A SAT-based framework for efficient constrained clustering." *Proceedings of the 2010 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 2010.

- Jabbour, Said, et al. "A SAT-Based Framework for Overlapping Community Detection in Networks." *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, Cham, 2017.

- Gilpin, Sean, and Ian Davidson. "Incorporating SAT solvers into hierarchical clustering algorithms: an efficient and flexible approach." *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011.

- Métivier, Jean-Philippe, et al. "Constrained clustering using SAT." *Advances in Intelligent Data Analysis XI* (2012): 207-218.

References – CP and Clustering

- Dao Bich, Duong, Khanh-Chuong, and Christel Vrain. "A declarative framework for constrained clustering." *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, Berlin, Heidelberg, 2013.

- Dao Bich, Duong, Khanh-Chuong, and Christel Vrain. "Constrained clustering by constraint programming." *Artificial Intelligence* (2015).

- Dao Bich, Duong Khanh-Chuong, and Vrain Christel. Constrained minimum sum of squares clustering by constraint programming. In Principles and Practice of Constraint Programming - 21st International Conference, CP 2015, Cork, Ireland, August 31 - September 4, 2015, Proceedings, pages 557–573, 2015.

- Dao Bich, Duong Khanh-Chuong, and Vrain Christel. "A filtering algorithm for constrained clustering with within-cluster sum of dissimilarities criterion." *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*. IEEE, 2013.

- Chabert Maxime and Solnon Christine. Constraint Programming for Multi-criteria Conceptual Clustering. In In Principles and Practice of Constraint Programming CP 2017.

- Tias et' al. "Repetitive branch-and-bound using constraint programming for constrained minimum sum-of-squares clustering." *ECAI 2016: 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague.*

References – MIP and Clustering

- Babaki, Behrouz, Tias Guns, and Siegfried Nijssen. "Constrained clustering using column generation." *International Conference on AI and OR Techniques in Constriant Programming for Combinatorial Optimization Problems*. Springer, Cham, 2014.

- Gilpin, Sean, Siegfried Nijssen, and Ian Davidson. "Formalizing hierarchical clustering as integer linear programming." *Proceedings of the twenty-seventh AAAI conference on artificial intelligence*. AAAI Press, 2013.

- Gilpin, Sean, and Ian Davidson. "A flexible ILP formulation for hierarchical clustering." *Artificial Intelligence Journal* (2015).

- Mueller, M. and Kramer, S. (2010). Integer Linear Programming Models for Constrained Clustering. In Proceedings of the 13th International Conference on Discovery Science, pages 159–173.

- Ouali, A., Loudni, S., Lebbah, Y., Boizumault, P., Zimmermann, A., and Loukil, L. (2016). Efficiently finding conceptual clustering models with integer linear programming. In IJCAI’16, New York, NY, USA, 9-15 July 2016, pages 647–654.

References – Applications and Novel Problems

- Kuo, Chia-Tung, and Ian Davidson. "A Framework for Outlier Description Using Constraint Programming." *AAAI*. 2016.

- Kuo, C. T., Ravi, S. S., Thi-Bich-Hanh Dao, Vrain, C., & Davidson, I. (2017, February). A Framework for Minimal Clustering Modification via Constraint Programming. In *AAAI* (pp. 1389-1395).

- Kotthoff, Lars, et al. "Complex Clustering Using Constraint Programming: Modeling Electoral Map." (2015).

- Kotthoff, Lars, et al. "Find your way back: mobility profile mining with constraints." *International Conference on Principles and Practice of Constraint Programming*. Springer, Cham, 2015.

Wrapping up

Why SAT/CP/ILP for ML and DM?

- SAT/CP/MIP can alleviate classical ML and DM limitations:
  1. Complex constraints
  2. Multiple (arbitrary) objectives
  3. Find one, multiple or all solutions
  4. Stay close to original data properties
  5. Solve discrete optimization problem directly

→ Allows solving more complex problems,
by building on existing tools

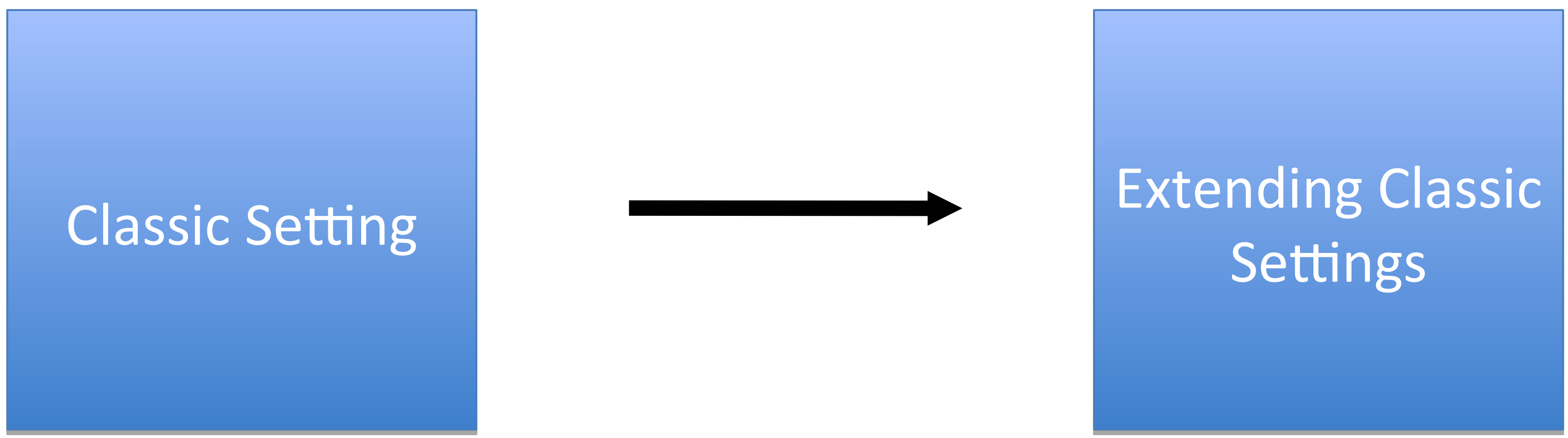
How Are SAT/MIP/CP Solvers Being Used in ML/DM Settings?

Classic Setting

Extending Classic Settings

Completely New Settings

Common theme (1/2)



Model classic problem to support additional constraints

- constraint-based itemset mining
- constraint-based sequence mining
- constrained hierarchical and centroid clustering
- constraint-based decision tree induction

Common theme (2/2)

Completely New Settings

- Novel problem variants and applications
  - constrained correlated itemset mining
  - direct mining of subj. interesting (relational) patterns
  - overlapping hierarchical clustering
  - exact multi-objective clustering
- New problems
  - dominance programming (multi-obj, relevant, ...)
  - outlier description problem
  - cluster modification
  - fairness in decision trees and other variations

Common theme (2/2)

Completely New Settings

And many using SAT/MIP/CP we did not have time to discuss:

- cis-regulatory module detection (itemset variant)
- toxiphore detection (set of itemset variant)
- trajectory mining
- Bayesian structure learning
- inverse frequent itemset mining

New problems:

- electoral map construction
- ranked tiling
- maximum order preserving submatrix
- pattern-guided k-anonimity

How Are SAT/MIP/CP Solvers Being Used in ML/DM Settings?

Pure SAT/MIP/CP Formulation

Extensions To SAT/MIP/CP Solvers

Hybrid Formulations

How to choose between SAT/MIP/CP?

No free lunch!

General guidelines:

- if decision problem: try SAT first
- if inherently Boolean: try (max)SAT first
- if few constraints or natural to relax: try MIP first
- if highly complex constraints: try CP first

Common theme

<u>Hybridize</u> state-of-the-art with solvers

- specialised algorithms in solver (extentions to solvers)
  - itemset mining coverage
  - correlated itemsets non-linear objective
  - relational patterns non-linear objective
  - sequence mining coverage
  - clustering objective functions
- chaining solvers with algorithms
  - pattern set mining
  - conceptual clustering
  - minimal cluster modificiation

Suitability of CP/MIP/SAT

- Problems we discussed typically were:
  - Discrete problems
  - With additional, complex, side-constraints
  - Worthwhile to enumerate all solutions or find a near-optimal one
- And main challenges we discussed:
  - generality vs efficiency trade-offs
  - how the problem is modeled matters
  - exploiting problem structure in the solution method

New problems

- distributed learning $\leftrightarrow$ DCOP
  (Distributed Constrained Optimisation)
- pattern sets
- streaming data / online setting
- learning under structural constraints
- explainable AI
- human-in-the-loop approaches:
  constraints to express knowledge

Future works

- Generic frameworks for complex ML and DM
  - for prototype and *baseline* approaches
  - can become part of applied ML and DM toolbox
- Hybridization
  - best of both worlds approaches
- New applications and problems
  - requiring complex constraints, discrete structure, etc

Data Mining and Machine Learning with CP/SAT/MIP

Slides at:
https://sites.uclouvain.be/cp4dm/tutorial/ijcai17/

Thank you for listening!
Questions and discussion welcome!

Siegfried Nijssen
Université catholique de Louvain, Belgium
siegfried.nijssen@uclouvain.be

Tias Guns
VUB Brussel and KU Leuven, Belgium
tias.guns@vub.be

Ian Davidson
University of California – Davis
davidson@cs.ucdavis.edu