Constraint solving

“Combinatorial problem = Model + Solve”

**Model** = specification of constraints over variables

**Solve** = search for satisfying/optimal solutions
- generic handling of variables and constraints
- efficient propagation of individual constraints

---

![Diagram](attachment:image.png)

Model → Translation → SOLVER → Output

- Model
- Data
- Translation
- SOLVER
- Output
Data Mining

“Extracting useful information from data”

- Machine learning: extracting predictive models
- Clustering: extracting meaningful groups
- Pattern mining: extracting regularities
- Recommender systems: extracting preferences
Solving data mining problems

Specific methods/algorithms for specific problems

Limited flexibility:

- New problems rarely fit existing methods well
- Tedious programming & hacks
- Refining solution methods is hard, but typical in the knowledge discovery cycle
Constraint solving can help!

- Complex constraints
- Reusing solving technology
- Adding/removing (user) constraints
- Exhaustive, optimal

Used in a wide range of problems in data mining
Active research directions

- **Pattern Mining**  P. Boizumault, B. Cremilleux, L. De Raedt, T. Guns, S. Jabbour, M. Jarvisalo, S. Loudni, S. Nijssen, B. O'Sullivan, W. Ugarte, ...

- **Clustering**  B. Babaki, I. Davidson, T.B.H. Dao, O. du Merle, S. Gilpin, P. Hansen, S. Nijssen, C. Vrain, ...

- **Structure learning**  C. Bessiere, J. Cussens, O. Grinchtein, M. Heule, T. Jaakkola, M. Meila, B. O'Sullivan, D. Sontag, N. Piterman, S. Verwer, ...
Pattern Mining

“Finding regularities in data”

Text Mining

Well, there’s egg and spam; egg sausage and spam; spam and bacon; egg bacon and spam; egg bacon sausage and spam; spam bacon sausage and spam; spam egg spam bacon and spam; spam sausage spam bacon spam tomato and spam; spam spam spam egg and spam; spam spam spam spam baked beans spam spam spam spam; or Lobster Thermidor, a Crevette with a mornay sauce served in a Provencale manner with shallots and aubergines garnished with truffle pate, brandy and spam.
Constraint-based Itemset Mining

- Fundamental enumeration problem
- Well studied
- Many constraints
- Many applications
Frequent Itemset Mining

**Find:** set of *items* appearing frequently (enumeration)

\[
\text{cover}(\{\text{Data Mining}, \text{Data Mining}\}, \{\text{Professor}, \text{Student}\}) = \{\text{Professor}, \text{Student}\}
\]

\[
\text{frequency}(\{\text{Data Mining}, \text{Data Mining}\}, \{\text{Professor}, \text{Student}\}) = \mid \{\text{Professor}, \text{Student}\} \mid = 2
\]
### CP for Itemset Mining

#### Coverage:
\[
\forall T_t: \quad T_t = 1 \iff \text{set}(I_1, \ldots, I_n) \subseteq \text{set(\text{row}_t)}
\]

#### Frequency:
\[
\sum_t T_t \geq Freq
\]
CP for Itemset Mining

Coverage:
\[ \forall I_i \quad I_i = 1 \Rightarrow \sum_t T_t D_{ti} \geq \text{Freq} \]

Frequency:
\[ \forall T_t \quad T_t = 1 \Leftrightarrow \sum_i I_i (1 - D_{ti}) = 0 \]

[L. De Raedt, T. Guns, S. Nijssen, KDD 2008]
CP4IM, basic model (MiniZinc)

% params/data
int: MinFreq;
int: NrI; int: NrT;
array [1..NrT,1..NrI] of 0..1: TDB;

% vars
array[1..NrI] of var 0..1: Items;
array[1..NrT] of var 0..1: Trans;

% Trans covered if Itemset subset of the transaction
constraint forall(t in 1..NrT) (  
    Trans[t] == 1 <-> sum(i in 1..NrI) (Items[i]*(1-TDB[t,i])) = 0 );

% each Item must be covered by sufficiently many Trans
constraint forall(i in 1..NrI) (  
    Items[i] == 1 -> sum(t in 1..NrT) (Trans[t]*TDB[t,i]) >= MinFreq );

[L. De Raedt, T. Guns, S. Nijssen, KDD 2008]
More constraints

- Coverage (required)
- Frequent
- Maximal
- Closed
- Delta-closed

Additional constraints:

\[ T_t = 1 \iff \sum_i I_i (1 - D_{ti}) = 0 \]

\[ I_i = 1 \Rightarrow \sum_t T_t D_{ti} \geq \text{Freq} \]

\[ I_i = 1 \iff \sum_t T_t D_{ti} \geq \text{Freq} \]

\[ I_i = 1 \iff \sum_t T_t (1 - D_{ti}) = 0 \]

\[ I_i = 1 \iff \sum_t T_t (1 - \delta - D_{ti}) = 0 \]

+ combinations!

[De Raedt, T. Guns, S. Nijssen, KDD 2008]
## Generality

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum frequency</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Maximum frequency</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Emerging patterns</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Condensed Representations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maximal</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Closed</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>$\delta$-Closed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Constraints on syntax</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max/Min total cost</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum average cost</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Max/Min size</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Constraints on labelled data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Minimum correlation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Maximum correlation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Basic Frequent Itemset Mining

CP (Gecode)

Minimum support

Specialised systems

coverage+frequency
Constraint-based Itemset Mining

![Graph showing runtime vs. MaxAvgCost for different systems.](image)

- **Specialised systems**
- **CP (Gecode)**

Legend:
- FIM_CP_1%
- FIM_CP_5%
- FIM_CP_10%
- PATTER_1%
- PATTER_5%
- PATTER_10%
- LCM_10%
Take away message 1.

Constraint Programming for Itemset Mining:
• Intuitive, reasonably compact encoding
• Generic: many constraints can be expressed
• Effective in case of tight constraints

Many extensions (not in this talk):
- Pattern set mining [M. Khiari, P. Boizumault, B. Cremilleux, ISMIS 2011]
  [T. Guns, S. Nijssen, L. De Raedt, TKDE 2013]
- Skypatterns / multi-objective [A. Soulet, C. Raïssi, M. Plantevit, B. Crémilleux, ICDM 2011]
  [W. Ugarte, P. Boizumault, S. Loudni, B. Cremilleux, ECAI 2014]
- SAT, BDD, ASP solvers [JP. Metivier, P. Boizumault, B. Cremilleux, M. Khiari, S. Loudni, IDA 2012]
  [H. Cambazard, T. Hadzi, B. O'Sullivan, ECAI 2010]
  [M. Jarvisalo, LPNMR 2011]
So far only enumeration problems.

What about optimisation problems?
Optimisation: discriminative itemsets

Known as: correlated itemset mining, discriminative itemset mining, contrast set mining, emerging itemsets, subgroup discovery, ...

- **Given**: labelled transactions

- **Find**: the itemset that *best correlates* with the class label
  - : \{+, \times\}
  - : \{+, +\}
Correlation function

\[ f(\sum_{t \in P} T_t, \sum_{t \in N} T_t) \geq \Theta \]

Example functions: chi^2, information gain, accuracy, ...

Common property: convex and zero-on-the-diagonal

- Existing pruning technique:
  only uses upper-bound of \( \sum T \)

- Our CP-based propagator:
  uses upper- and lower-bound of \( \sum T \)
  and look-ahead formulation
  \[ I_i = 1 \Rightarrow \ldots \]

much stronger propagation!

[T. Guns, S. Nijssen, L. De Raedt, AIJ 2012]
## Correlated itemset mining

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

Runtime in seconds

- -: out of memory
- >: time-out

[T. Guns, S. Nijssen, L. De Raedt, AIJ 2012]
Take away message 2.

Correlated itemset mining:

- (constrained) optimisation problem
- Novel propagator for correlation functions
- CP encoding+propagator beats state-of-the-art

Open issues:

- Identify other complex functions with good properties?
- Statistical evaluation of patterns common; statistical constraints in CP?
- Potential for interactive classifier construction?

[R. Rossi, S. Prestwich, S. Armagan Tarim, ECAI 2014]
What about scalability?
Overhead in CP or fundamental difference?

Developed specialised constraint solver:

- Only constraints used in CP4IM (reified linear Boolean sums)
- Inspiration from itemset miners:
  - array of Booleans: 2 bitvectors, for upper- and lowerbound
  - never copy the data (propagator has views on the data)
  - very fine-grained propagator activation

```
Miner search tree

{}  
   /\  
  / \  
{a}  {b}  
   /\   /\  
  / \  / \  
{a,b} {}  {}  

CP search tree

{?,?}  
 /\  
/ \  
{?,0} {?,1}  
   /\   /\  
  / \  / \  
{0,0} {1,0} {0,1} {1,1}  
    /\    /\    /\    /\  
   ={}  ={a}  ={b}  ={a,b}
```

[S. Nijssen, T. Guns, ECMLPKDD 2010]
Specialised solver

Old, Gecode

New, DMCP
Frequent Itemset Mining, scalability

Old, Gecode

T10I4D100K (Frequent)

New, DMCP
Take away message 3.

Scalability:
- Out-of-the-box solvers: generality/efficiency trade-off
- Overhead in implementation, not in methodology

General lessons for CP?
- Fine-grained propagation (AC-5, events etc) Half-reification  
  [T. Feydy, Z. Somogyi, P.J. Stuckey, CP 2011]
- Bitvectors for arrays of Booleans  [L. Michel, P. Van Hentenryck, CP 2012]
- Scaling up CP solvers to large data?  
  [J-G Fages. X. Lorca, T. Petit, ECAI 2014]
Constraint Solving = Model + Solve

Encoding problem in effective constraints

(Itemset Mining: reified linear sums)

- Mining algorithms
- Generic CP solvers
- Specialised CP solvers

Better modeling support for data mining?
A modeling language for pattern mining?

Goals:
- High-level, natural notation (for data miners)
- User-defined constraints
- Efficient solving (with\&without extra constraints)
- Support existing mining algorithms?

Observation: can formulate as CSP, reuse high-level CSP language?
A modeling language for pattern mining?

MiningZinc

- Based on the established MiniZinc language
  - High-level mathematical-like notation
  - User-defined constraints and functions
  - Solver independent (10+ CP solvers & SAT & MIP)

- **Modeling**: Pattern mining specific constraints and functions

- **Solving**: automatic model reformulations; Can detect and use specialised mining algorithms (LCM, Eclat, FPgrowth)

include "lib_itemsetmining.mzn"

int: NrI; int: NrT; int: MinFreq;
array[1..NrT] of set of int: TDB;

var set of 1..NrI: Items;

constraint card(cover(Items, TDB)) >= MinFreq;

array [1..NrI] of int: Cost;
int: MinCost;

constraint sum(i in Items) (Cost[i]) >= MinCost

solve satisfy;
Solving: standard MiniZinc

- model.mzn
- data.dzn
- mzn2fzn
- FZN solver
  - gecode
  - CPX
  - or-tools
  - SCIP
  - ...

RAW TEXT
Solving: MiningZinc (IJCAI'13)

1) Flatten (partial)
2) Detect DM algo

model.mzn -> mngzn2fzn -> lib_im.mzn

FZN solver or DM algorithm

data.dzn
Example matching rule:

\[ \{ T = \text{cover}(I, D), ST = \text{card}(T), ST \geq V \} \rightarrow \text{LCM}(+V, +D, -I) \]
Take away message 4.

Modeling:
- Can build on existing high-level languages
- Solver independence:
  - Automatic model reformulation
  - Novel combinations of CP/DM algorithms

Open questions
- Multiple execution strategies: algorithm selection?
- Problems not fitting standard CP
  - Skyline patterns / multi-objective [W. Ugarte, P. Boizumault, S. Loudni, B. Cremilleux, ECAI 2014]
Active research directions

- Pattern Mining
- Clustering: constraint-based clustering
- Structure learning
Clustering: grouping by similarity
Hierarchical clustering

Surveyed as most popular method in practice

General algorithm:
- Initially every instance is separate cluster
- Repeat
  - Compute similarities between clusters
  - Merge two most similar clusters
Hierarchical clustering

Constraints:
- Must-link and cannot-link
- Conditional ML/CL
- Precedence
- Constraints on levels, ...

SAT-based modification (Horn-SAT!):
- Initially every instance is separate cluster
- Repeat
  - Compute similarities between clusters
  - Do until merge found:
    - Query SAT solver: can merge the two most similar clusters?

[S. Gilpin, I. Davidson, KDD 2011]
Hierarchical clustering

Still a local, greedy algorithm...
Can the global optimum be found?

Using Integer Linear Programming:
- Variables $M_{ij}$: what level points $i$ & $j$ are merged at
- Constraints: reflexivity, symmetry, transitivity

$$\arg \max_{M,O,Z} \sum_{a,b,c \in \text{Instances}} w_{abc} \times O_{abc}$$

subject to:
- $O, Z, M$ are integers.
- $0 \leq O \leq 1$, $0 \leq Z \leq 1$
- $1 \leq M \leq L$
- $-L \leq M(a,c) - M(a,b) - (L+1)O_{abc} \leq 0$
- $-L \leq M(a,b) - M(a,c) - (L+1)Z_{ab \geq ac} + 1 \leq 0$
- $-L \leq M(b,c) - M(a,c) - (L+1)Z_{bc \geq ac} + 1 \leq 0$
- $Z_{ab \geq ac} + Z_{bc \geq ac} \geq 1$

- Objective: ultrametric favoring early joins of similar instances

[S. Gilpin, S. Nijssen, I. Davidson, AAAI 2013]
Hierarchical clustering

ILP formulation: finds optimal solution

- Good for high-value data
- More robust to noise than greedy:

Novel settings:

- relaxing transitivity constraints = hierarchy with overlapping clusters

[S. Gilpin, S. Nijssen, I. Davidson, AAAI 2013]
Take away message 5.

Constraint-based clustering:
- constraint solving already supports complex constraints

Optimal clustering:
- value of computation time vs solution quality
- robustness against local minima/noise

Other clustering settings:
- Solving: constrained-based K-means-like, spectral clustering, ...
Active research directions

- Pattern Mining
- Clustering
- Structure learning
Structure learning

Motivation:

- Constraint Solving typically effective when problem has structure
- Reformulate problem as known (graph) problem (e.g. coloring) + reuse existing solving technology
- Enforce certain graph properties (tree, directed, acyclic)
- Can find optimal solutions
Structure learning

- Decision tree induction
  - Smallest tree with perfect training accuracy
  - Global 'tree' constraint in CP
  - Better testing accuracy than existing, unpruned, methods

- Automaton construction
  - Typical: build acceptance tree, merge states consistently
  - Merging can be seen as coloring under conditions
    → formulate as graph coloring using CP/SAT
  - + more compact SAT encoding, symm. breaking, redundant constr.
  - Scalable hybrid: start from greedily constructed partial solution

[C. Bessiere, E. Hebrard, B. O'Sullivan, CP 2009]

[O. Grinchtein, M. Leucker, N. Piterman, IJCAR 2006]

[SAT [M. Heule, S. Verwer, ICGI 2010]]
Structure learning

Bayesian network structure learning

- Each node has candidate parent sets (of bounded size)
- Additive (decomposable) quality measure over parent sets
  - Weighted MaxSAT
    - Hard constraints: has parent + acyclicity,
    - Soft constraints: score of parent set
  - ILP / Cutting planes (acyclicity)

Tutorial: [J. Cussens, B. Malone, C. Yuan, IJCAI 2013]

[J. Cussens, UAI 2008]
[T. Jaakkola, D. Sontag, A. Globerson, M. Meila, AISTATS 2010]
[J. Cussens, UAI 2011]
Take away message 6.

Structure learning:
- Complex constraints: graph problems common in constraint solving
- Reuse of efficient solving technology: can beat state-of-the-art with proper encoding
- Scalability – hybridization:
  - Heuristically construct partial network
  - Find optimal completion

Open questions:
- Other structure learning problems
- Necessary conditions? (bounded size, decomposable functions, ...)
- Improving modeling, encoding, solving
Overview

Modeling paradigm already prevalent in A.I.: SAT, ILP/MIP, Graphical Models, STRIPS, ...

Increased use in data mining too:

- On wide range of tasks
- For handling complex constraints (declarative)
- Reuse of solving technology: can beat state-of-the-art

Creates shift from 'programming' to 'modelling'

→ efficient encoding of problem is crucial
Challenges

Big data vs Smart data vs Valuable data

Nevertheless, scalability:
- Novel encodings or propagators
- Solver technology/search strategy
- Hybridization
- Generic heuristic search?

→ new benchmarks for solvers...

Other tasks: plenty more in pattern mining, clustering, struct. learning

Beyond CSPs? multi-objective, soft constraints, preferences, statistics
Thank you for listening

Constraint Solving

Data Mining

Questions?

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