Towards Cross-Fertilization Between Data Mining and Constraints

Lakhdar Saïs

CRIL - CNRS UMR 8188
Université d'Artois, France

Joint work with
Said Jabbour, Badran Raddaoui, Yakoub Salhi, Karim Tabia
Takeaki Uno

http://www.cril.univ-artois.fr/decMining/

Mini Symposium DECADE, Lyon, november 9th 2017

Data mining

Extracting **useful knowledge** from data

Mainly driven by real-life applications including biology (e.g. gene expression data), business intelligence (e.g. market basket), Web (e.g. XML data), ...

At the crossroad of many disciplines:

- Databases,
- Artificial Intelligence,
- Statistics
- and data analysis, combinatorics, algorithmic, ...

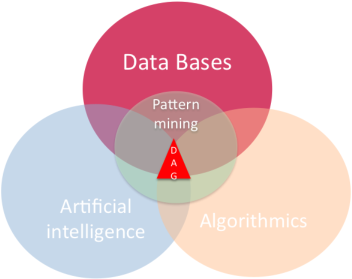
Data mining

Extracting **useful knowledge** from data

Several research issues:

- **Pattern mining**
- **Clustering & community detection**
- Machine learning
- Recommander systems
- ...

ANR Défis - DAG (2009 - 2013)
Declarative Approaches for Enumerating Interesting Patterns



`http://liris.cnrs.fr/dag/`

- CRIL, University d'Artois, Lens
- LIMOS, University of Blaise Pascal, Clermont-Ferrand
- LIRIS, University Claude Bernard, Lyon 1, Lyon

Outline

Part I: Declarative approaches for pattern mining problems

Sequence, Itemset and association rules mining

Part II: Data mining $\leftarrow$ AI

Symmetries in Itemset Mining

Part III: Data mining $\rightarrow$ AI

Mining-based Compression Approach of Propositional Formulae

Conclusion & Perspectives

Boolean Satisfiability Problem (SAT)

- A conjunction of clauses:

$$\overbrace{(x_1 \vee \cdots \vee x_l)}^{\textit{clause}} \wedge (y_1 \vee \cdots \vee y_m) \wedge (z_1 \vee \cdots \vee z_n) \cdots$$

- Clause: a disjunction of literals (x, $\neg x$)

- Example :

$$\Phi = \underbrace{\overbrace{(p \vee \neg q \vee \neg r)}^{1}}_{\textit{horn}} \wedge \overbrace{(p \vee \neg q \vee s)}^{1} \wedge \underbrace{\overbrace{p}^{1}}_{\textit{unary}} \wedge \underbrace{\overbrace{(r \vee \neg s)}^{1}}_{\textit{binary}}$$

$\mathcal{M}(p) = 1$ and $\mathcal{M}(r) = 1$ (Model)

Satisfiability: $\exists \mathcal{M}, \mathcal{M}(\Phi) = 1$ (NP-complete [Cook 71])

Boolean Satisfiability Problem (SAT)

- Spectacular progress $\rightarrow$ Modern SAT solvers
  - application instances with millions of variables and clauses
- Many applications
  - Formal Verification
  - Planning
  - Bioinformatics
  - Cryptography
  - ...
- CRIL Projects
  - Microsoft Research Cambridge (UK): 2008-2012
- CRIL Solvers : Glucose (Sequential), ManySAT (Parallel)
- Books:
  - Lakhdar Sais (eds), Problème SAT : Progrès et Défis, Hermes Publishing Ltd, pp.352, 2008
  - Youssef Hamadi and Lakhdar Sais (eds), Handbook of Parallel Constraint Reasoning, Springer, February 2018

Part I : SAT based approach for Sequences mining

- Alphabet: a set Σ
- Wildcard (or Joker): $\circ \notin \Sigma$
- Sequence S: word $S_1 S_2 \ldots S_n$ in Σ^*
- Pattern P: word $P_1 P_2 \ldots P_m$ in $(\Sigma \cup \{\circ\})^*$
  - $P_1 \neq \circ$ and $P_m \neq \circ$
  - Sequences are patterns

$abbac$, $ab \circ c \circ \circ d$, ~~$\circ ab \circ c$~~, ~~$ab \circ c \circ$~~

Frequent Patterns in a Sequence

Let $P = P_1 P_2 \ldots P_m$ and $P' = P'_1 P'_2 \ldots P'_n$

- $P \subseteq_p P'$ if $\forall i \in \{1, \ldots, m\}$:
  - either $P_i = P'_{p+i-1}$
  - or $P_i = \circ$
- $P \subseteq P'$ if $\exists p$ st. $P \subseteq_p P'$
- $L_S(P) = \{p \mid P \subseteq_p S\}$

Example

$a \circ b \subseteq_2 a\underline{aab}baabab$ $\quad L_{aaabbaabab}(a \circ b) = \{2, 3, 6\}$

$a \circ \circ b \subseteq_1 \underline{a \circ ab} \circ b$ $\quad a \circ \circ b \subseteq_3 a \circ \underline{ab \circ b}$

$a \circ bb \not\subseteq a \circ \circ b$

Definition (Finding frequent patterns in a sequence)

Input: a sequence S and an integer λ

Output: all patterns P st. $|L_S(P)| \geqslant \lambda$

Representing the Searched Pattern as Boolean Variables

Pattern $P = P_1 P_2 \ldots P_m$
For each position i: $1 \leqslant i \leqslant m$:

- For each character $a \in \Sigma \cup \{\circ\}$
  - variable p_i^a is true iff $P_i = a$

$$\neg p_1^{\circ} \quad \wedge \quad \bigwedge_{i=1}^{m} \bigvee_{a \in \Sigma \cup \{\circ\}} p_i^a \quad \wedge \quad \bigwedge_{i=1}^{m} \bigwedge_{a,b \in \Sigma \cup \{\circ\}, a \neq b} (\neg p_i^a \vee \neg p_i^b) \tag{1}$$

Set trailing $p_{m'+1}^{\circ} \ldots p_m^{\circ}$ to true to express pattern of size $m' < m$

Location and Support

The pattern P is found at position k in $S = S_1 \dots S_n$:

$$loc(k, P, S) = \bigwedge_{i=1}^{m} (p_i^{\circ} \vee p_i^{S_{i+k-1}})$$

Assuming $S_{i+k-1} = \circ$ when $i + k - 1 > n$

New variables $t_1 \dots t_n$, encoding $L_S(P)$

- t_k is true iff $P \subseteq_k S$

$$supp(P, S) = \bigwedge_{i=1}^{n} (t_k \Leftrightarrow loc(k, P, S)) \quad (2)$$

Example

Sequence:

$$a \quad a \quad a \quad b \quad b \quad a \quad a \quad b \quad a \quad b$$

Pattern (max size 6):

$$a \quad \circ \quad b \quad \circ \quad \circ \quad \circ$$

Example

Sequence:

$a \quad a \quad a \quad b \quad b \quad a \quad a \quad b \quad a \quad b$

Pattern (max size 6):

a	$\circ$	b	$\circ$	$\circ$	$\circ$
p_1^a	p_2^a	p_3^a	p_4^a	p_5^a	p_6^a
p_1^b	p_2^b	p_3^b	p_4^b	p_5^b	p_6^b
p_1°	p_2°	p_3°	p_4°	p_5°	p_6°

true, false

Example

Sequence:

a	a	a	b	b	a	a	b	a	b
t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}

$$L_S(P) = \{2, 3, 6\}$$

Pattern (max size 6):

a	$\circ$	b	$\circ$	$\circ$	$\circ$
p_1^a	p_2^a	p_3^a	p_4^a	p_5^a	p_6^a
p_1^b	p_2^b	p_3^b	p_4^b	p_5^b	p_6^b
p_1°	p_2°	p_3°	p_4°	p_5°	p_6°

true, false

Frequency constraint

$$freq(P, S, \lambda) = \sum_{i=1}^{n} t_k \geqslant \lambda \tag{3}$$

Several possible encodings of the boolean cardinality constraint:

- Transformation of 0/1 linear inequalities to CNF [Warners 1996]
- Cardinality networks [Asín et al. 2011]
- BDD encoding [Bailleux at al. 2003]

Polynomial Encoding of $\sum_{j=1}^{n} x_j \geqslant \lambda$ to CNF

$$\bigwedge_{k=1}^{\lambda} (\neg p_{ki} \vee x_i), \quad i = 1, \ldots, n \tag{4}$$

$$\bigvee_{i=1}^{n} p_{ki}, \quad k = 1, \ldots, \lambda \tag{5}$$

$$\bigwedge_{1 \leqslant k < k' \leqslant \lambda} (\neg p_{ki} \vee \neg p_{k'i}), \quad i = 1, \ldots, n \tag{6}$$

(5) and (6) encode the pigeon hole problem PHP_n^{λ}

- p_{ki} expresses that pigeon k is in hole i
- x_i is true if the hole i contains one of the pigeons k for $k = 1, \ldots, \lambda$

Complexity $O(\lambda \times n)$ vars and $O(n \times \lambda^2)$ clauses
With Symmetry breaking $\Rightarrow O(\lambda \times (n - \lambda))$ vars and clauses

Part I: SAT based approach for Itemsets Mining

- Transactions database $\mathcal{D}$ over a set of items $\mathcal{I} = \{Camus, Djaout, Djebar, Kateb, \ldots, Mimouni\}$

$\mathcal{T}_{id}(\mathcal{D})$	itemset
000	*Djebar*, *Djaout*, *Dib*
001	*Feraoun*, *Mimouni*, *Kateb*
002	*Djebar*, *Dib*
003	*Camus*, *Mimouni*, *Kateb*
004	*Fanon*, *Haddad*
005	*Mimouni*, *Mammeri*

- Support: $\mathcal{S}(\{Mimouni, Kateb\}, \mathcal{D}) = |\{001, 003\}| = 2$

Frequent Itemset Mining problem

Compute $\mathcal{FIM}(\mathcal{D}, \lambda) = \{I \subseteq \mathcal{I} \mid \mathcal{S}(I, \mathcal{D}) \geqslant \lambda\}$

Example

$\mathcal{FIM}(\mathcal{D}, 2) =$
$\{\{Mimouni\}, \{Kateb\}, \{Mimouni, Kateb\}, \{Djebar\},$
$\{Dib\}, \{Djebar, Dib\}\}$

Condensed Representations of Frequent Itemsets

Maximal frequent

$$Max(\mathcal{D}, \lambda) = \{I \in \mathcal{FIM}(\mathcal{D}, \lambda) | \forall J \supset I, J \notin \mathcal{FIM}(\mathcal{D}, \lambda)\}$$

Closed frequent itemsets

$$Cl(\mathcal{D}, \lambda) = \{I \in \mathcal{FIM}(\mathcal{D}, \lambda) | \forall J \supset I, \mathcal{S}(J, \mathcal{D}) < \mathcal{S}(I, \mathcal{D})\}$$

Example

$$Max(\mathcal{D}, 2) = \{\{Djebar, Dib\}, \{Mimouni, Kateb\}\}$$
$$Cl(\mathcal{D}, 2) = \{\{Mimouni\}, \{Djebar, Dib\}, \{Mimouni, Kateb\}\}$$

Problem Statement

Mining Frequent Closed itemsets $\mathcal{FCIM}_\lambda$

- **Input**: $\mathcal{D} = \{(0, t_0), \ldots, (n-1, t_{n-1})\}$ a transaction database over a set of items $\mathcal{I}$. λ a minimum support threshold.
- **Output**: all frequent closed itemsets

SAT-based Encoding for $\mathcal{FCIM}_\lambda$

- Associate to each item $a \in \mathcal{I}$ a boolean variable p_a.
  - Such boolean variables encode the candidate itemset $I \subseteq \mathcal{I}$, i.e., $p_a = true$ **iff** $a \in I$.
- $\forall\, i \in \{0, \ldots, n-1\}$, associate to the *i-th* transaction a Boolean variable b_i.

SAT-based Encoding for $\mathcal{FCIM}_\lambda$

A constraint to capture all the transactions where the candidate itemset does not appear:

$$\bigwedge_{i=0}^{n-1} (b_i \leftrightarrow \bigvee_{a \in \mathcal{I} \setminus t_i} p_a) \tag{7}$$

A constraint to force the candidate itemset to be **closed**:

$$\bigwedge_{a \in \mathcal{I}} (\bigwedge_{i=0}^{n-1} \overline{b_i} \rightarrow a \in t_i) \rightarrow p_a \tag{8}$$

A constraint to consider only the frequent itemsets:

$$\sum_{i \in 0 \ldots n-1} \overline{b_i} \geqslant \lambda \tag{9}$$

Note: for association rules and variants see our [IJCAI'2016, PAKDD'2017] papers

Outline

Part I: Declarative approaches for pattern mining problems

- Sequence, Itemset and association rules mining

Part II: Data mining ← AI

- Symmetries in Itemset Mining

Part III: Data mining → AI

- Mining-based Compression Approach of Propositional Formulae

Conclusion & Perspectives

Itemset mining

Output of huge size

- Difficult to retrieve useful information.
- Reducing the size of the output is crucial for practical data mining
  - Search for condensed representations by exploiting the structure of the itemsets data
    (e.g. closed, maximal, discriminative itemset patterns, etc.)

Symmetries

- Fundamental concept (structural knowledge) in Computer Science, Mathematics, Physics and many other domains.
  - Many human artifacts (e.g. classroom in a university, aircraft seats, circuit patterns) and entities in nature (e.g. plants, molecules, DNA sequences, atoms) exhibits symmetries.
  - $\Rightarrow$ Useful for reasoning and understanding more complex entities and systems.

Symmetries in CP and SAT

- Symmetry resolution proof system [Krishnamurthy 1985]
- Dynamic symmetry detection and elimination in propositional calculus [Benhamou & Saïs 1992]
- Symmetry breaking predicates [Crawford 1992]
- Variable and value symmetries [Puget 1993]
- Many other contributions [Sakallah 2011, Walsh 2012...]

How to exploit symmetries in itemset mining?

1. by dynamic integration in Apriori-like algorithms for search space pruning.
2. by rewriting the transaction databases in a preprocessing step (items elimination).
   - $\rightarrow$ new transaction database + symmetry group.
   - $\rightarrow$ condensed representation of the output.

Symmetry in Frequent Itemset Mining

Definition (Transaction Renaming)

A renaming f over $\mathcal{T}_{id}(\mathcal{D})$ is a bijective mapping from $\mathcal{T}_{id}(\mathcal{D})$ to $\mathcal{T}_{id}(\mathcal{D})$.

We can extend a renaming f to $\mathcal{D}$ as follows:
$f(\mathcal{D}) = \{(f(t_i), I) | (t_i, I) \in \mathcal{D}\}$.

Definition (Permutation)

A permutation σ over $\mathcal{I}$ is a bijective mapping from $\mathcal{I}$ to $\mathcal{I}$.

We extend a permutation σ to $\mathcal{D}$ as follows:
$\sigma(\mathcal{D}) = \{(t_i, \sigma(I)) | (t_i, I) \in \mathcal{D}\}$ where $\sigma(I) = \{\sigma(a) | a \in I\}$.

Symmetry in Frequent Itemset Mining

Each permutation σ can be represented by a set of cycles $c_1 \ldots c_n$ where each cycle $c_i = (a_1, \ldots, a_k)$ is a list of elements of $\mathcal{I}$ such that $\sigma(a_j) = a_{j+1}$ for $j = 1, \ldots, k-1$, and $\sigma(a_k) = a_1$.

Definition (Symmetry)

A symmetry of $\mathcal{D}$ is a permutation $\sigma \in \mathcal{P}(\mathcal{I})$ such that there exists a transaction renaming f over $\mathcal{T}_{id}(\mathcal{D})$ where $\sigma(\mathcal{D}) = f(\mathcal{D})$ i.e. $f^{-1}(\sigma(\mathcal{D})) = \mathcal{D}$.

Proposition

Let σ a symmetry of $\mathcal{D}$, λ a minimal support threshold and I an itemset. $I \in \mathcal{FIM}(\mathcal{D}, \lambda)$ iff $\sigma(I) \in \mathcal{FIM}(\mathcal{D}, \lambda)$.

Symmetry in Frequent Itemset Mining

Example

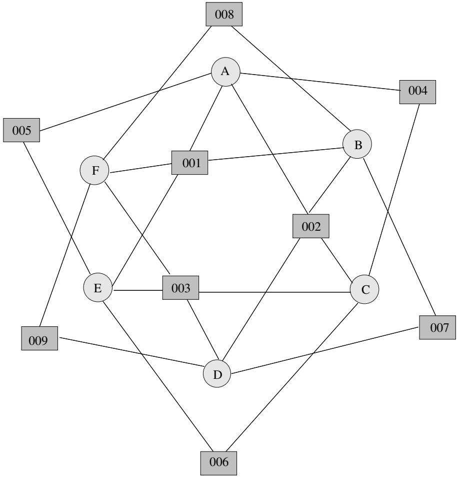
σ = (C,E)(D,F) is a symmetry

t_i	itemset			
001	A,	B,	E,	F
002	A,	B,	C,	D
003	C,	D,	E,	F
004	A,	C,		
005	A,	E,		
006	C,	E,		
007	B,	D,		
008	B,	F,		
009	D,	F,		

$$f(t_i) = \begin{cases} 001 & \text{if } t_i\text{=002} \\ 002 & \text{if } t_i\text{=001} \\ 003 & \text{if } t_i\text{=003} \\ 004 & \text{if } t_i\text{=005} \\ 005 & \text{if } t_i\text{=004} \\ 006 & \text{if } t_i\text{=006} \\ 007 & \text{if } t_i\text{=008} \\ 008 & \text{if } t_i\text{=007} \\ 009 & \text{if } t_i\text{=009} \end{cases}$$

Symmetry Detection in Transaction Databases

- Convert the original problem $\mathcal{D}$ into a colored undirected graph $\mathcal{G}$, where vertices are labeled with colors.
- Look for the automorphism group of $\mathcal{G}$.
- Symmetries of $\mathcal{D}$ are equivalent to the automorphisms of the colored undirected graph $\mathcal{G}$.
- Employ a general-purpose graph symmetry tool to uncover the symmetries [Mckay'81, Aloul'03].



t_i	itemset			
001	A,	B,	E,	F,
002	A,	B,	C,	D
003	C,	D,	E,	F
004	A,	C		
005	A,	E		
006	C,	E		
007	B,	D		
008	B,	F		
009	D,	F		

Symmetry Pruning

Integration in Apriori-like algorithm

$\rightarrow$ proceeds by a level-wise search of the elements of $\mathcal{FIM}(\mathcal{D}, \lambda)$.

1. Starts by computing the elements of $\mathcal{FIM}(\mathcal{D}, \lambda)$ of size 1.
2. Assuming $\mathcal{FIM}(\mathcal{D}, \lambda)$ of size n known, computes a set of candidates of size $n + 1$ so that I is a candidate if and only if all its subsets are in $\mathcal{FIM}(\mathcal{D}, \lambda)$.
3. This procedure is iterated until no more candidate is found.

Symmetry-Based Pruning in Apriori-like algos

- Let $\mathcal{D}$ be a transaction database such that $\mathcal{I}(\mathcal{D}) = \{A, B, C, D\}$ and σ is a symmetry such that σ=$(A, D)(B, C)$.
- Assume that the itemsets $\{A\}$, $\{B\}$, $\{C\}$ and $\{D\}$ are frequent. We also assume that in iteration 2, we find that the itemset $\{A, B\}$ is not frequent.

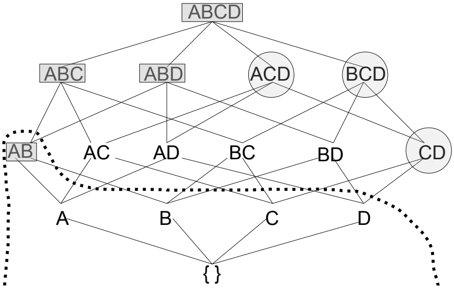


Figure: Symmetry Pruning

Symmetry Breaking

- Breaking symmetries in a preprocessing step.
  - Eliminate items from the original transaction database.
  - The frequent itemsets generated using the new transaction database together with the symmetry group can be used to retrieve the whole set of frequent itemsets of the original

Symmetry Breaking

Let $\mathcal{D}$ a transaction database and $\sigma = (a, b)(c, d)$ a symmetry

$\mathcal{FIM}(\mathcal{D}, \lambda)$=$\{a, \dots\}$, $\{b, \dots\}$, $\{c, \dots\}$, $\{d, \dots\}$, $\{a, b, \dots\}$, $\{a, c, \dots\}$, $\{a, d, \dots\}$, $\{b, c, \dots\}$, $\{b, d, \dots\}$, $\{c, d, \dots\}$

$$\begin{aligned} \{a, \dots\} &\rightarrow \{b, \dots\} \\ \{a, d, \dots\} &\rightarrow \{b, c, \dots\} \\ \{a, c, \dots\} &\rightarrow \{b, d, \dots\} \\ \{a, b, \dots\} &\rightarrow \{a, b, \dots\} \end{aligned}$$

$$\begin{aligned} \{b, \dots\} &\rightarrow \{a, \dots\} \\ \{b, c, \dots\} &\rightarrow \{a, d, \dots\} \\ \{d, \dots\} &\rightarrow \{c, \dots\} \\ \{b, d, \dots\} &\rightarrow \{a, c, \dots\} \end{aligned}$$

Symmetry Breaking

Let $\mathcal{D}$ a transaction database and $\sigma = (a, b)(c, d)$ a symmetry

$\mathcal{FIM}(\mathcal{D}, \lambda)$=$\{a, \dots\}$, $\cancel{\{b, \dots\}}$, $\{c, \dots\}$, $\cancel{\{d, \dots\}}$, $\{a, b, \dots\}$,
$\{a, c, \dots\}$, $\{a, d, \dots\}$, $\cancel{\{b, c, \dots\}}$, $\cancel{\{b, d, \dots\}}$, $\{c, d, \dots\}$

$\{a, \dots\}$	$\rightarrow$	$\{b, \dots\}$	$\{b, \dots\}$	$\rightarrow$	$\{a, \dots\}$
$\{a, d, \dots\}$	$\rightarrow$	$\{b, c, \dots\}$	$\{b, c, \dots\}$	$\rightarrow$	$\{a, d, \dots\}$
$\{a, c, \dots\}$	$\rightarrow$	$\{b, d, \dots\}$	$\{d, \dots\}$	$\rightarrow$	$\{c, \dots\}$
$\{a, b, \dots\}$	$\rightarrow$	$\{a, b, \dots\}$	$\{b, d, \dots\}$	$\rightarrow$	$\{a, c, \dots\}$

Symmetry Breaking

Let $\mathcal{D}$ a transaction database and $\sigma = (a, b)(c, d)$ a symmetry

$\mathcal{FIM}(\mathcal{D}, \lambda)$=$\{a, \dots\}$, ~~$\{b, \dots\}$~~, $\{c, \dots\}$, ~~$\{d, \dots\}$~~, $\{a, b, \dots\}$, $\{a, c, \dots\}$, $\{a, d, \dots\}$, ~~$\{b, c, \dots\}$~~, ~~$\{b, d, \dots\}$~~, $\{c, d, \dots\}$ + σ

$\{a, \dots\}$	$\rightarrow$	$\{b, \dots\}$	$\{b, \dots\}$	$\rightarrow$	$\{a, \dots\}$
$\{a, d, \dots\}$	$\rightarrow$	$\{b, c, \dots\}$	$\{b, c, \dots\}$	$\rightarrow$	$\{a, d, \dots\}$
$\{a, c, \dots\}$	$\rightarrow$	$\{b, d, \dots\}$	$\{d, \dots\}$	$\rightarrow$	$\{c, \dots\}$
$\{a, b, \dots\}$	$\rightarrow$	$\{a, b, \dots\}$	$\{b, d, \dots\}$	$\rightarrow$	$\{a, c, \dots\}$

- $\Rightarrow$ b can be removed from each $T \in \mathcal{D}$ if $\{a, b\} \not\subset T$
- $\Rightarrow$ d can be removed from each $T \in \mathcal{D}$ if $\{a, d\} \not\subset T$ and $\{c, d\} \not\subset T$

Symmetry Breaking

Proposition

Let $\mathcal{D}$ a transaction database and

$\sigma = (x_1, y_1)(x_2, y_2)\cdots(x_j, y_j)\cdots(x_n, y_n)$ a symmetry

$\Rightarrow$ y_j can be removed from each $T \in \mathcal{D}$ if $\{x_i, y_j\} \not\subset T, \forall i \leqslant j$

Remark

Symmetries can be broken independently

Symmetry Breaking: an example

t_i	itemset			
001	A,	B,	E,	F,
002	A,	B,	C,	D
003	C,	D,	E,	F
004	A,	C		
005	A,	E		
006	C,	E		
007	B,	D		
008	B,	F		
009	D,	F		

σ_1 = (A C)(B, D)

σ_2 = (A B)(C, D) (E F)

σ_3 = (C,E)(D,F)

t_i	itemset			
001	A,	B,	~~E~~,	~~F~~
002	A,	B,	C,	D
003	~~C~~	~~D~~	~~E~~	~~F~~
004	A,	C,		
005	A,	~~E~~,		
006	~~C~~	~~E~~		
007	~~B~~	~~D~~		
008	~~B~~	~~F~~		
009	~~D~~	~~F~~		

Table: Itempair-based Symmetry Breaking approach

Symmetry Breaking: an example

t_i	itemset			
001	A,	B,	E,	F,
002	A,	B,	C,	D
003	C,	D,	E,	F
004	A,	C		
005	A,	E		
006	C,	E		
007	B,	D		
008	B,	F		
009	D,	F		

σ_1 = (A C)(B, D)
σ_2 = (A B)(C, D) (E F)
σ_3 = (C,E)(D,F)

t_i	itemset			
001	A,	B,		
002	A,	B,	C,	D
003				
004	A,	C		
005	A			
006				
007				
008				
009				

Table: Itempair-based Symmetry Breaking approach

Outline

Motivation

Growing success obtained in solving real-world SAT problems highlights a real transition to industrial and commercial scale.

- increasing use of SAT technology to solve new real-world applications (bioinformatics, cryptography, etc.)
- a rapid growth in the size of the CNF instances encoding real-world problems.

$\rightarrow$ **Challenge**: Design of new efficient models for representing and solving SAT instances of very large sizes (”Big” instances).

Modeling in SAT

- Knowledge representation using CNF formulae

		6	1		2	5		
	3	9				1	4	
				4				
9		2		3		4		1
	8						7	
1		3		6		8		9
				1				
	5	4				9	1	
		7	5		3	2		

8	4	6	1	7	2	5	9	3
1	3	9	6	5	8	1	4	2
5	2	1	3	4	9	7	6	8
9	6	2	8	3	7	4	5	1
4	8	5	9	2	1	3	7	6
1	7	3	4	6	5	8	2	9
2	9	8	7	1	4	6	3	5
3	5	4	2	8	6	9	1	7
6	1	7	5	9	3	2	8	4

- Example : $n \times n$ Sudoku
  - Associate to each cell, n propositional variables
  - Each cell contains at least one value:

    $\bigwedge_{l=1}^{n} \bigwedge_{c=1}^{n} (\bigvee_{v=1}^{n} p_{(l,c,v)}) \implies n^2$ clauses of size n
- Leads usually to formulae of huge size

Modeling in SAT: an example from formal verification

Name of the CNF instance : post-cbmc-zfcp-2.8-u2.cnf (BMC)
p cnf **11 483 525** (vars) **32 697 150** (clauses)

1 -3 0
2 -3 0
1 -2 3 0

$x_3 = x_1 \wedge x_2$

⋮ 1million pages later

-11482897 -11483041 -11483523 0
11482897 11483041 -11483523 0
11482897 -11483041 11483523 0
-11482897 11483041 11483523 0

$x_3 \leftrightarrow x_4 \leftrightarrow x_5$

-11483518 -11483524 0
-11483519 -11483524 0
-11483520 -11483524 0
-11483521 -11483524 0
-11483522 -11483524 0
-11483523 -11483524 0
11483518 11483519 11483520 11483521 11483522 11483523 11483524 0

$x_6 = (x_7 \wedge x_8 \wedge x_9 \wedge x_{10} \wedge x_{11} \wedge x_{12})$

-8590303 -11483524 -11483525 0
8590303 11483524 -11483525 0
8590303 -11483524 11483525 0
-8590303 11483524 11483525 0

$x_{13} \leftrightarrow x_{14} \leftrightarrow x_{15}$

-11483525 0

Transformation - Extension principle [G. Tseitin 1965]

- Introduce new variables to represent truth value of sub-formulae
- Example : DNF $\longrightarrow$ CNF

$$(x_1 \wedge y_1) \vee (x_2 \wedge y_2) \vee \cdots \vee (x_n \wedge y_n)$$

- Naïve approach: 2^n clauses and $n \times 2^n$ literals
  $(x_1 \vee \cdots \vee x_{n-1} \vee x_n) \wedge (x_1 \vee \cdots \vee x_{n-1} \vee y_n) \wedge \cdots \wedge (y_1 \vee \cdots \vee y_{n-1} \vee y_n)$
- Tseitin approach: $2 \times n + 1$ clauses and $n + 2 \times 2 \times n$ literals
  $(z_1 \vee \cdots \vee z_n) \wedge (\neg z_1 \vee x_1) \wedge (\neg z_1 \vee y_1) \wedge \cdots \wedge (\neg z_n \vee x_n) \wedge (\neg z_n \vee y_n)$

CNF formula as transactions database

- Goals : Reduce
  - *the size of the Formula* : reduce the number of literals using the frequent sets of literals and Tseitin extension principle
  - *the solving time* (bonus)
- Items: literals
- Transactions: clauses > 2

Example

$(x_1 \vee \neg x_2 \vee \neg x_3) \wedge (x_1 \vee \neg x_2 \vee x_4) \wedge x_1 \wedge (x_3 \vee \neg x_4)$

itemset
$x_1, \neg x_2, \neg x_3$
$x_1, \neg x_2, x_4$

Reduce the number of literals

- Introduce new Boolean variables:

$$(x_1 \vee \cdots \vee x_n \vee \alpha_1) \wedge \cdots \wedge (x_1 \vee \cdots \vee x_n \vee \alpha_k)$$

$$\underset{\Longrightarrow}{\textit{equivalent} \text{ w.r.t. } \textit{SAT}}$$

$$(y \vee \alpha_1) \wedge \cdots \wedge (y \vee \alpha_k) \wedge (\neg y \vee x_1 \vee \cdots \vee x_n)$$

- $n \times k$ literals substituted by $k + n + 1$ literals
- Size reduction: $n \times k - (k + n + 1) > 0 \rightarrow k > \frac{n+1}{n-1}$
- Minimum support threshold: $k \begin{cases} \geqslant 4 & \text{si } n = 2 \\ \geqslant 3 & \text{si } n = 3 \\ \geqslant 2 & \text{otherwise} \end{cases}$

Closed Vs. Maximal

- Maximal $\subseteq$ Closed : more informations with closed

$$(x_1 \vee \ldots \vee x_k \vee \ldots \vee x_n \vee \alpha_1) \wedge \cdots \wedge (x_1 \vee \ldots \vee x_k \vee \ldots \vee x_n \vee \alpha_m) \wedge$$
$$(x_1 \vee \ldots \vee x_k \vee \beta_1) \wedge \cdots \wedge (x_1 \vee \ldots \vee x_k \vee \beta_{m'})$$

We suppose that the set of itemsets are frequent

$\Rightarrow$ Max $= \{\{x_1, \ldots, x_n\}\}$, Clos $= \{\{x_1, \ldots, x_k\}, \{x_1, \ldots, x_n\}\}$

- Use of $\{x_1, \ldots, x_n\}$:

$$(y \vee \alpha_1) \wedge \cdots \wedge (y \vee \alpha_m) \wedge$$
$$(x_1 \vee \ldots \vee x_k \vee \beta_1) \wedge \cdots \wedge (x_1 \vee \ldots \vee x_k \vee \beta_{m'}) \wedge$$
$$(\neg y \vee x_1 \vee \ldots \vee x_n)$$

- Use of $\{x_1, \ldots, x_k\}$:

$$(y \vee \alpha_1) \wedge \cdots \wedge (y \vee \alpha_m) \wedge$$
$$(z \vee \beta_1) \wedge \cdots \wedge (z \vee \beta_{m'}) \wedge$$
$$(\neg y \vee z \vee x_{k+1} \vee \ldots \vee x_n) \wedge (\neg z \vee x_1 \vee \ldots \vee x_k)$$

Weighted Patterns

- The best:
  - X if $|X| \times \mathcal{S}(X) - (\mathcal{S}(X) + |X| + 1) \geqslant |Y| \times \mathcal{S}(Y) - (\mathcal{S}(Y) + |Y| + 1)$
  - Y otherwise
- Associates a weight to frequent itemsets:

$$|X| \times \mathcal{S}(X) - (\mathcal{S}(X) + |X| + 1)$$

Overlaps

- Problem with overlaps:
  - $\{x_1, x_2, x_3\}$ et $\{x_2, x_3, x_4\}$ two frequents itemsets s.t. $\mathcal{S}(\{x_1, x_2, x_3\}) = 3$, $\mathcal{S}(\{x_2, x_3, x_4\}) = 3$ and $\mathcal{S}(\{x_1, x_2, x_3, x_4\}) = 2$
  - Use of $\{x_1, x_2, x_3\} \longrightarrow \mathcal{S}(\{x_2, x_3, x_4\}) = 1$
- Overlap classes:
  - X overlaps with Y ($X \sim Y$): $X \cap Y \neq \emptyset$
  - overlaps class (Overlap class): an equivalence class (transitive closure of $\sim$)

$$Y \in [X] \ \ \textit{iff} \ \ Y = Y_1 \sim Y_2 \sim \ldots \sim Y_k = X$$

  - Overlap class = Connected Component on $G = (V, E)$,
    - V the set of patterns $\mathcal{P}$
    - $E = \{\{P_i, P_j\} | P_i \cap P_j \neq \emptyset\}$.
  - Optimal solution $\longrightarrow$ optimal solution in each overlaps class

Compression as an Optimisation Problem

The compression problem can be formulated as an optimisation problem

Problem : $\mathcal{C}omp(\Phi, \mathcal{P})$

- **Input**: Φ a CNF formula, and $\mathcal{P}$ a set of patterns
- **Output**: a compressed formula Φ of minimal size using $\mathcal{P}$

Compression as an Optimisation Problem

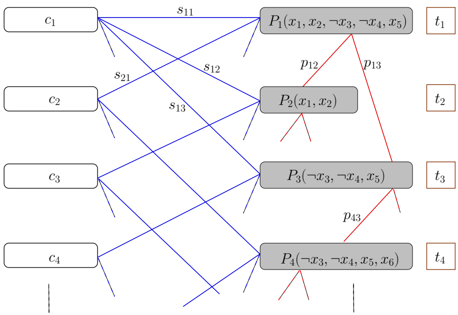
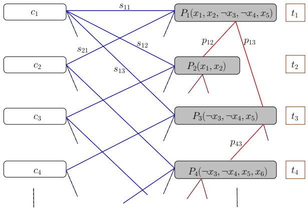


Figure: Compression using location problem

- $\mathcal{S} = \{s_{ij} | P_j \subseteq C_i\}$. If $subst(c_i, P_j)$, then $s_{ij} = 1$; else $s_{ij} = 0$.
- $\mathcal{T} = \{t_j | 1 \leqslant j \leqslant m\}$. If $used(P_j)$, then $t_j = 1$, else $t_j = 0$
- $\mathcal{P} = \{p_{ij} | P_j \subseteq P_i\}$. If $subst(P_i, P_j)$, then $p_{ij} = 1$; else $p_{ij} = 0$.

Compression as an Optimisation Problem



A **formulation** as 0/1 linear program

$$Max \sum_{s_{ij} \in \mathcal{S}} (|P_j| - 1) \times s_{ij} + \sum_{p_{ij} \in \mathcal{P}} (|P_j| - 1) \times p_{ij} - (\sum_{j=1}^{m} (|P_j| + 1) \times t_j$$

1. $s_{ij} - t_j \leqslant 0 \qquad s_{ij} \in \mathcal{S}$
2. $p_{ij} - t_j \leqslant 0,\ p_{ij} - t_i \leqslant 0 \qquad p_{ij} \in \mathcal{P}$
3. $s_{ij} + s_{ik} \leqslant 1 \qquad s_{ij} \in \mathcal{S}, s_{ik} \in \mathcal{S}, P_j \cap P_k \neq \emptyset$
4. $s_{ij} \in \{0, 1\} \qquad s_{ij} \in \mathcal{S}$
5. $t_j \in \{0, 1\} \qquad 1 \leqslant j \leqslant m$

Gready Algorithm

- Search for frequent closed patterns (sub-clauses)
- Sort the patterns according to their weights (size reduction)
- Substitution of the patterns following the ordering

Algorithm

Require: A formula ϕ, an overlap class of closed frequent itemsets C

while $C \neq \emptyset$ **do**
    $I \leftarrow C.MostIntersting Element()$;
    $\phi.replace(I, y)$;
    $\phi.Add(I, y)$;
    $C.remove(I)$;
    $C.replaceSubset(I, y)$;
    $C.removeUninterestingElements()$;
    $C.updateSupports()$;
end while
return ϕ

Experiments: Industrial SAT instances

Instance	orig.	comp.	% red
1dlx_c_iq57_a	190 Mb	164 Mb	13.68 %
6pipe_6_ooo.*-as.sat03-413	11 Mb	7.7 Mb	30.00 %
9dlx_vliw_at_b_iq6.*-*04-347	76 Mb	65 Mb	14.47 %
abb313GPIA-9-c.*.sat04-317	21 Mb	6.9 Mb	67.14 %
E05F18	3.7 Mb	2.2 Mb	40.54 %
eq.atree.braun.11.unsat	120 Kb	72 Kb	40.00 %
eq.atree.braun.12.unsat	144 Kb	88 Kb	38.88 %
k2mul.miter.*-as.sat03-355	1.5 Mb	1.3 Mb	13.33 %
korf-15	1.2 Mb	752 Kb	37.33 %
rbcl_xits_08_UNSAT	1.1 Mb	856 Kb	22.18 %
SAT_dat.k45	3.5 Mb	2.6 Mb	25.71 %
traffic_b_unsat	18 Mb	12 Mb	33.33 %
x1mul.miter.*-as.sat03-359	1.1 Mb	928 Kb	15.63 %
9dlx_vliw_at_b_iq3	19 Mb	15 Mb	21.05 %
9dlx_vliw_at_b_iq4	31 Mb	26 Mb	16.12 %
AProVE07-09	2.8 Mb	2.7 Mb	3.57 %
eq.atree.braun.10.unsat	96 Kb	56 Kb	41.66 %
goldb-heqc-frg1mul	348 Kb	328 Kb	5.74 %
minand128	7.7 Mb	2.6 Mb	66.23 %
ndhf_xits_09_UNSAT	2.6 Mb	2.1 Mb	19.23 %
velev-pipe-o-uns-1.1-6	5.5 Mb	4.4 Mb	20.00 %

Table: Results of Mining4SAT : a general approach

Application: A compact representation of 2-CNF

instance	#cls	#bin	(%) bin
velev-pipe-o-uns-1.1-6	304026	268354	88,26 %
9dlx_vliw_at_b_iq2	542253	500227	92,24 %
1dlx_c_iq57_a	8562505	7567948	88,38 %
7pipe_k	751116	722278	96,16 %
SAT_dat.k100.debugged	670701	523153	78,00 %
BM_FV_2004_rule_batch	445444	339588	76,23 %
sokoban-sequential-p145-*.040-*	1413816	1364160	96,48 %
openstacks-*-p30_1.085-*	1621926	1601145	98,71 %
aaai10-planning-ipc5-*-12-step16	1029036	991140	96,31 %
k2fix_gr_rcs_w8.shuffled	271393	270136	99,53 %
homer17.shuffled	1742	1716	98,50 %
gripper13u.shuffled-as.sat03-395	38965	35984	92,34 %
grid-strips-grid-y-3.045-*	2750755	2695230	97,98 %

Table: Ratio of binary clauses in some SAT instances

Application: A compact representation of 2-CNF

Example

Let us consider the following 2-CNF Φ:

$$\begin{aligned}\Phi = \quad & (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_1 \vee x_4) \wedge (x_1 \vee x_5) \quad \wedge \\ & (x_1 \vee x_6) \wedge (x_1 \vee x_7) \wedge (x_2 \vee x_3) \wedge (x_2 \vee x_4) \quad \wedge \\ & (x_2 \vee x_5) \wedge (x_2 \vee x_6) \wedge (x_2 \vee x_7) \wedge (x_3 \vee x_4) \quad \wedge \\ & (x_3 \vee x_6) \wedge (x_3 \vee x_7) \wedge (x_3 \vee x_5) \wedge (x_4 \vee x_5) \quad \wedge \\ & (x_4 \vee x_6) \wedge (x_4 \vee x_7) \wedge (x_5 \vee x_6) \wedge (x_5 \vee x_7) \quad \wedge \\ & (x_6 \vee x_7)\end{aligned}$$

Definition (B-implication)

A *B-implication* is a Boolean formula of the following from : $x \vee \beta(x)$ where $\beta(x)$ is a conjunction of literals.

Application: A compact representation of 2-CNF

Using the complete order relation $x_1 \prec \ldots \prec x_7$ over $\mathcal{L}_\Phi$ rewrite Φ as set of B-implications $B^1_{[\vee(\wedge)]}(\Phi)$:

$$\{[x_1 \vee (x_2 \wedge x_3 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7)],$$
$$[x_2 \vee (x_3 \wedge x_4 \wedge x_5 \wedge x_6 \wedge x_7)],$$
$$[x_3 \vee (x_4 \wedge x_5 \wedge x_6 \wedge x_7)],$$
$$[x_5 \vee (x_6 \wedge x_7)],$$
$$[x_6 \vee (x_7)]\}$$

tid	itemset					
tid_{x_1}	x_2	x_3	x_4	x_5	x_6	x_7
tid_{x_2}		x_3	x_4	x_5	x_6	x_7
tid_{x_3}			x_4	x_5	x_6	x_7
tid_{x_4}				x_5	x_6	x_7
tid_{x_5}					x_6	x_7
tid_{x_6}						x_7

Application: A compact representation of sets of 2-CNF

FIM process on the conjunctive part of $B^1_{\vee[\wedge]}(\Phi)$

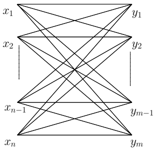
Using $\{x_5, x_6, x_7\}$ a 4-frequent itemset, we can rewrite $B^1_{[\vee(\wedge)]}(\Phi)$ as:

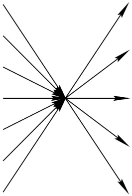
$$B^2_{\vee[\wedge]}(\Phi) = \begin{array}{l} \{[x_1 \vee (x_2 \wedge x_3 \wedge y)]\ , \\ [x_2 \vee (x_3 \wedge x_4 \wedge y)]\ , \\ [x_3 \vee (x_4 \wedge y)]\ , \\ [x_5 \vee (x_6 \wedge x_7)]\ , \\ [x_6 \vee (x_7)]\ , \\ [\neg y \vee (x_5 \wedge x_6 \wedge x_7)]\} \end{array}$$

$\mathrm{CNF}(B^2_{[\vee(\wedge)]}(\Phi)) =$

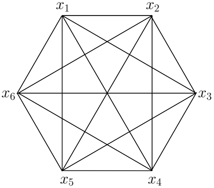
$$\begin{array}{ll} (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_1 \vee y) & \wedge \\ (x_2 \vee x_3) \wedge (x_2 \vee x_4) \wedge (x_2 \vee y) & \wedge \\ (x_3 \vee x_4) \wedge (x_3 \vee y) & \wedge \\ (x_5 \vee x_6) \wedge (x_5 \vee x_7) & \wedge \\ (x_6 \vee x_7) & \wedge \\ (\neg y \vee x_5) \wedge (\neg y \vee x_6) \wedge (\neg y \vee x_7) & \end{array}$$

Two particular cases: bi-cliques and cliques





$n \times m$ clauses $\Rightarrow$ $n + m$ clauses and 1 new variable



$\mathcal{O}(n^2)$ clauses $\Rightarrow$ $\mathcal{O}(n)$ clauses and $\mathcal{O}(n)$ new variables $\Rightarrow$ $\sum_{i=1}^{n} x_i = 2$

More details on bi-cliques

Let $\Phi = [(x_1 \vee y_1) \wedge (x_1 \vee y_2) \wedge \cdots \wedge (x_1 \vee y_m)] \ldots [(x_n \vee y_1) \wedge (x_n \vee y_2) \wedge \cdots \wedge (x_n \vee y_m)]$

- Using a complete order relation defined by: $f(x_i) = i, f(y_j) = n + j$.
- $B_{[\vee(\wedge)]}(\Phi)$ corresponds exactly to $\{(x_i \vee [y_1 \wedge y_2 \wedge \cdots \wedge y_m]) | 1 \leqslant i \leqslant n\}$
- Using a single closed frequent itemset $\{y_1, y_2, \ldots, y_m\}$

$\Phi' = [\bigwedge_{1 \leqslant i \leqslant n}(x_i \vee z)] \wedge [\bigwedge_{1 \leqslant j \leqslant m}(\neg z \vee y_j)]$.

Experiments: Industrial SAT instances

Instance	orig.	comp.	% red
velev-pipe-o-uns-1.1-6	5.5 Mb	3.2 Mb	41.81 %
9dlx_vliw_at_b_iq2	11 Mb	6 Mb	44.45 %
1dlx_c_iq57_a	190 Mb	124 Mb	34.73 %
7pipe_k	14 Mb	5.4 Mb	61.42 %
SAT_dat.k100.debugged	16 Mb	13 Mb	18.75 %
IBM_FV_2004_rule_batch _2_31_1_SAT_dat.k80.debugged	9.7 Mb	7.5 Mb	22.68 %
sokoban-sequential-p145-*.040-*	24 Mb	14 Mb	41.66 %
openstacks-*-p30_1.085-*	30 Mb	26 Mb	13.33 %
aaai10-planning-ipc5-*-12-step16	17 Mb	12 Mb	29.41 %
k2fix_gr_rcs_w8.shuffled	3.4 Mb	1.7 Mb	50.00 %
homer17.shuffled	20 Kb	16 Kb	20.00 %
gripper13u.shuffled-as.sat03-395	524 Kb	364 Kb	30.35 %
grid-strips-grid-y-3.045-*	52 Mb	42 Mb	19.23 %

Table: Results of Mining4Binary: a 2-CNF approach

Combining Binary and Non Binary Clauses

$$\begin{array}{ccc} x_0 \vee \neg x_4, & x_0 \vee \neg x_5, & x_0 \vee \neg x_6, \\ \neg x_3 \vee \neg x_4, & \neg x_3 \vee \neg x_5, & \neg x_3 \vee \neg x_6, \end{array}$$

$$\begin{array}{rcl} \neg x_0 \vee x_1 & \vee & \boxed{x_4 \vee x_5 \vee x_6}, \\ x_3 & \vee & \boxed{x_4 \vee x_5 \vee x_6}, \\ \neg x_1 \vee x_2 & \vee & \boxed{x_4 \vee x_5 \vee x_6}, \\ \neg x_2 \vee x_3 & \vee & \boxed{x_4 \vee x_5 \vee x_6} \end{array}$$

Suppose that $(x_4 \vee x_5 \vee x_6)$ is frequent

Combining Binary and Non Binary Clauses

$$(x_0 \vee [\neg x_4 \wedge \neg x_5 \wedge \neg x_6])$$
$$(\neg x_3 \vee [\neg x_4 \wedge \neg x_5 \wedge \neg x_6])$$

$$\begin{array}{rcl}
\neg x_0 \vee x_1 & \vee & \left| \, x_4 \vee x_5 \vee x_6 \, \right|, \\
x_3 & \vee & \left| \, x_4 \vee x_5 \vee x_6 \, \right|, \\
\neg x_1 \vee x_2 & \vee & \left| \, x_4 \vee x_5 \vee x_6 \, \right|, \\
\neg x_2 \vee x_3 & \vee & \left| \, x_4 \vee x_5 \vee x_6 \, \right|
\end{array}$$

Combining Binary and Non Binary Clauses

$$(x_0 \vee \neg[x_4 \vee x_5 \vee x_6])$$
$$(\neg x_3 \vee \neg[x_4 \vee x_5 \vee x_6])$$

$$\begin{array}{rcl} \neg x_0 \vee x_1 & \vee & \left| x_4 \vee x_5 \vee x_6 \right|, \\ x_3 & \vee & \left| x_4 \vee x_5 \vee x_6 \right|, \\ \neg x_1 \vee x_2 & \vee & \left| x_4 \vee x_5 \vee x_6 \right|, \\ \neg x_2 \vee x_3 & \vee & \left| x_4 \vee x_5 \vee x_6 \right| \end{array}$$

Combining Binary and Non Binary Clauses

$$
\begin{array}{rcc}
x_0 \vee \neg \mathbf{y} & & \\
\neg x_3 \vee \neg \mathbf{y} & & \\
\neg x_0 \vee x_1 & \vee & \mathbf{y} \\
x_3 & \vee & \mathbf{y} \\
\neg x_1 \vee x_2 & \vee & \mathbf{y} \\
\neg x_2 \vee x_3 & \vee & \mathbf{y} \\
\neg \mathbf{y} & \vee & x_4 \vee x_5 \vee x_6 \\
\mathbf{y} \vee \neg x_4 & & \\
\mathbf{y} \vee \neg x_5 & & \\
\mathbf{y} \vee \neg x_6 & &
\end{array}
$$

Experiments

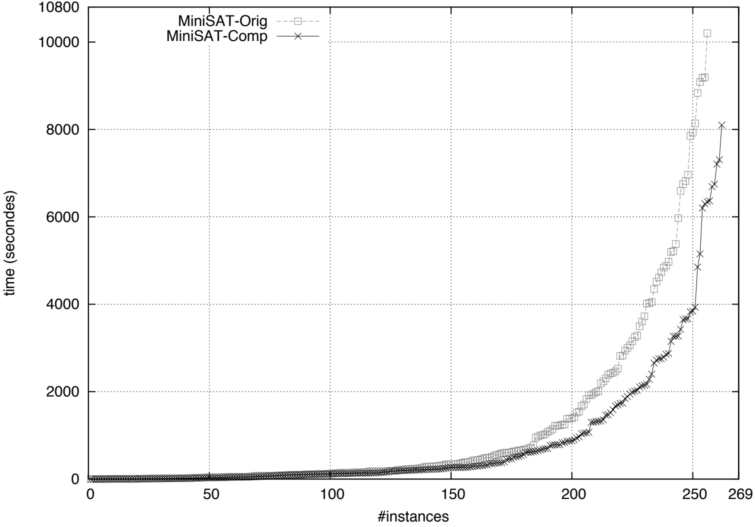


Figure: MiniSAT on SAT instances with and without compression

Application: A compact Graph Representation

For free, we can apply our approach for graphs.

- 2-CNF $\leftrightarrow$ graphs
- Adjacency lists $\leftrightarrow$ A set of B-implications
- $2 \rightarrow [4, 6, 8, 12] \leftrightarrow 2 \vee [4 \wedge 6 \wedge 8 \wedge 12]$

Outline

Part I: Declarative approaches for pattern mining problems

- Sequence, Itemset and association rules mining

Part II: Data mining $\leftarrow$ AI

- Symmetries in Itemset Mining

Part III: Data mining $\rightarrow$ AI

- Mining-based Compression Approach of Propositional Formulae

Conclusion & Perspectives

Perspectives

- Cross-fertilization between AI and Data mining
  - DM $\leftarrow$ AI (e.g. preferences, symmetries, knowledge compilation, etc.)
  - DM $\rightarrow$ AI (e.g. extracting structural knowledge, compression, etc.)
  - ...

Thank you for your attention