Approches Déclaratives pour la Fouille de Données

Lakhdar Saïs

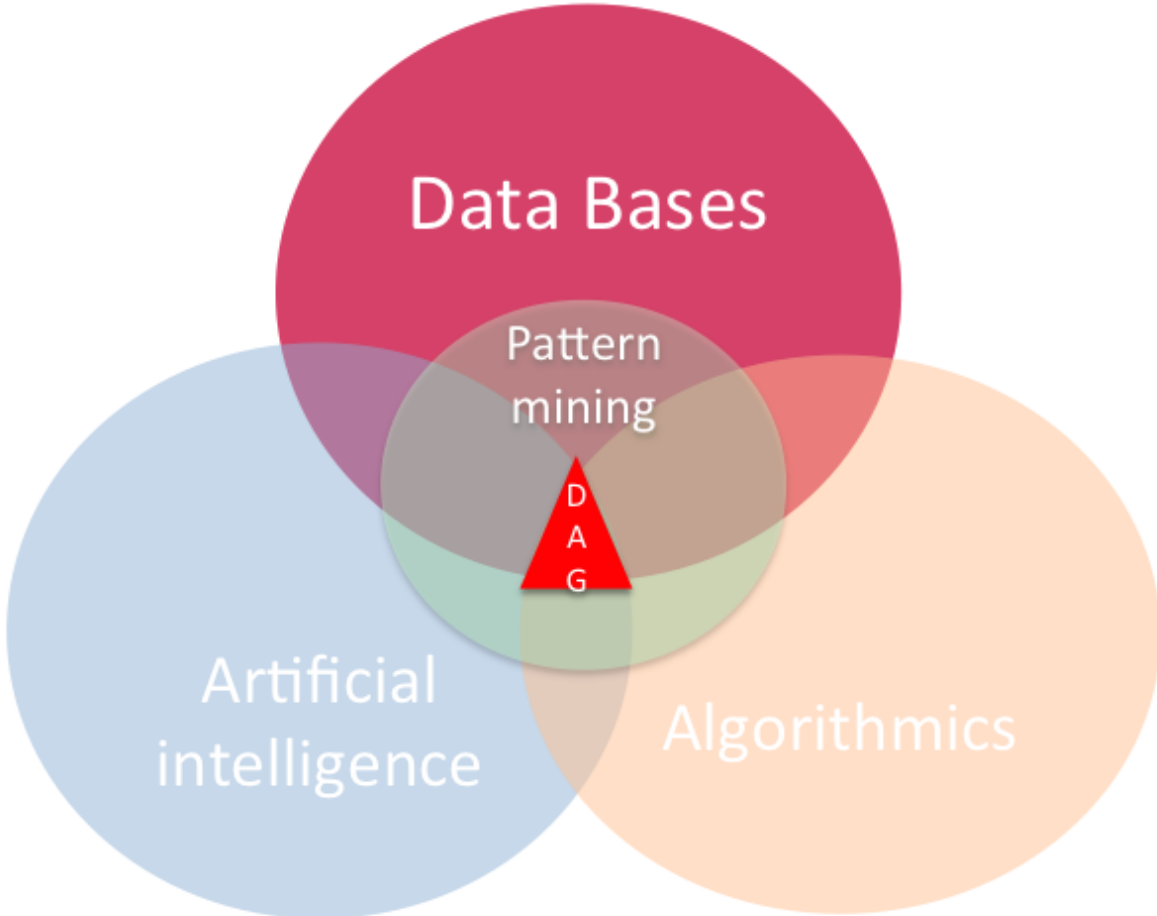
CRIL - CNRS UMR 8188
Université d'Artois, France

Work done in the framework of Project DAG ANR Défis 2009
Joint work with Emmanuel Coquery, Said Jabbour, Mehdi Khiari, Yakoub Salhi, Karim Tabia

February 20, 2014

Outline

DAG: 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

DAG: Objectives and Challenges

- Objectives
  - Cross-fertilization between artificial intelligence (CSP, SAT), combinatorial algorithmics and databases
  - Bringing original solutions to fundamental pattern mining problems
  - Definition of high level declarative languages for describing interesting pattern enumeration problems
- Two major challenges
  - Challenge 1: Definition of classes of problems for enumerating interesting patterns
  - Challenge 2: Design of declarative, efficient and generic pattern mining systems

Motivation

- Lot of contributions in pattern mining:
  - Different kind of patterns
  - Efficiency of enumeration algorithms
  - Various predicates
  - ...

- Most are dedicated approaches
  - Lack of declarativity

  $\Rightarrow$ Changing slightly the problem statement requires deep changes in implementations.

Some Results

1. A SAT-Based Approach for Discovering Frequent Closed and Maximal Patterns in a Sequence. E. Coquery, S. Jabbour, L. Sais, Y. Salhi [ECAI'12]

Some Results

1. A SAT-Based Approach for Discovering Frequent Closed and Maximal Patterns in a Sequence. E. Coquery, S. Jabbour, L. Sais, Y. Salhi [ECAI'12]
2. Boolean Satisfiability for sequence mining, S. Jabbour, L. Saïs, Y. Salhi [CIKM'13]

Some Results

1. A SAT-Based Approach for Discovering Frequent Closed and Maximal Patterns in a Sequence. E. Coquery, S. Jabbour, L. Sais, Y. Salhi [ECAI'12]
2. Boolean Satisfiability for sequence mining, S. Jabbour, L. Saïs, Y. Salhi [CIKM'13]
3. A Pigeon-Hole Based Encoding of Cardinality Constraints Jabbour, L. Sais et Y. Salhi [ICLP'13]

Some Results

1. A SAT-Based Approach for Discovering Frequent Closed and Maximal Patterns in a Sequence. E. Coquery, S. Jabbour, L. Sais, Y. Salhi [ECAI'12]
2. Boolean Satisfiability for sequence mining, S. Jabbour, L. Saïs, Y. Salhi [CIKM'13]
3. A Pigeon-Hole Based Encoding of Cardinality Constraints Jabbour, L. Sais et Y. Salhi [ICLP'13]
4. Symmetries in Itemset Mining. S. Jabbour, L. Sais, Y. Salhi, K. Tabia [ECAI'12]

Some Results

1. A SAT-Based Approach for Discovering Frequent Closed and Maximal Patterns in a Sequence. E. Coquery, S. Jabbour, L. Sais, Y. Salhi [ECAI'12]
2. Boolean Satisfiability for sequence mining, S. Jabbour, L. Saïs, Y. Salhi [CIKM'13]
3. A Pigeon-Hole Based Encoding of Cardinality Constraints Jabbour, L. Sais et Y. Salhi [ICLP'13]
4. Symmetries in Itemset Mining. S. Jabbour, L. Sais, Y. Salhi, K. Tabia [ECAI'12]
5. A mining-Based Approach for Size Reduction of CNF Formulae. S. Jabbour, L. Sais, Y. Salhi, T. Uno [CIKM'13]
6. Mining Top-k Frequent Closed Itemset Mining Using Top-k SAT Preferred Models. S. Jabbour, L. Sais and Y. Salhi [ECML-PKDD'13]

Some Results

1. A SAT-Based Approach for Discovering Frequent Closed and Maximal Patterns in a Sequence. E. Coquery, S. Jabbour, L. Sais, Y. Salhi [ECAI'12]
2. Boolean Satisfiability for sequence mining, S. Jabbour, L. Saïs, Y. Salhi [CIKM'13]
3. A Pigeon-Hole Based Encoding of Cardinality Constraints Jabbour, L. Sais et Y. Salhi [ICLP'13]
4. Symmetries in Itemset Mining. S. Jabbour, L. Sais, Y. Salhi, K. Tabia [ECAI'12]
5. A mining-Based Approach for Size Reduction of CNF Formulae. S. Jabbour, L. Sais, Y. Salhi, T. Uno [CIKM'13]
6. Mining Top-k Frequent Closed Itemset Mining Using Top-k SAT Preferred Models. S. Jabbour, L. Sais and Y. Salhi [ECML-PKDD'13]
7. Penalty-based Preferences in Itemset Mining. S. Jabbour, S. Kaci, L. Sais, Y. Salhi [Submitted]

Outline

Sequences and patterns

- Alphabet: a set of items Σ
- Wildcard (or Joker): $\circ \notin \Sigma$
- Sequence of items S: a word $S_0 S_1 \ldots S_{n-1}$ in Σ^*
- Pattern P: a word $P_0 P_1 \ldots P_{m-1}$ in $(\Sigma \cup \{\circ\})^*$
  - $P_0 \neq \circ$ and $P_{m-1} \neq \circ$
  - Sequences are patterns
  - Examples:

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

Inclusion

Let $P = P_0 P_1 \dots P_{m-1}$ and $P' = P'_0 P'_1 \dots P'_{n-1}$

- $P \preceq_l P'$ if $\forall i \in \{0, \dots, m-1\}$:
  - either $P_i = P'_{l+i}$
  - or $P_i = \circ$
- $P \preceq P'$ if $\exists l$ st. $P \preceq_l P'$
- $L_S(P) = \{l \mid P \preceq_l S\}$
- Shift: $L_S(P) + d = \{p + d \mid p \in L_S(P)\}$

$a \circ b \preceq_2 aaabbbaabab$

$a \circ \circ b \preceq_0 a \circ ab \circ b$

$a \circ \circ b \preceq_2 a \circ ab \circ b$

$a \circ bb \not\preceq a \circ \circ b$

$L_{aaabbbaabab}(a \circ b) = \{1, 2, 5\}$

$L_{aaabbbaabab}(a \circ b) + 3 = \{4, 5, 8\}$

Frequent Patterns

Definition (Finding frequent patterns in a sequence)

Input: a sequence S and a minimal support threshold λ
Output: all patterns P s.t. $|L_S(P)| \geqslant \lambda$

Property (Anti-monotonicity)

If $P \preceq P'$ then, for some d:

$$L_S(P') + d \subseteq L_S(P)$$

Closed Frequent Patterns

Definition (Finding closed frequent patterns)

Input: a sequence S and a minimal support threshold λ
Output: all patterns P s.t.:

- $|L_S(P)| \geqslant \lambda$
- there is no pattern Q and no integer d s.t.:
  - $P \prec Q$
  - $L_S(Q) + d = L_S(P)$

Maximal Frequent Patterns

Definition (Maximal frequent patterns)

Input: a sequence S and a minimal support threshold λ
Output: all patterns P st.:

- $|L_S(P)| \geqslant \lambda$
- there is no pattern Q s.t.:
  - $P \prec Q$
  - $|L_S(Q)| \geqslant \lambda$

Outline

An Encoding of Frequent Pattern Mining

- Boolean variables: for each a in Σ, we associate k_a Boolean variables

$$p_{a,0}, \ldots, p_{a,k_a-1}$$

where $k_a = min(max(L_S(a)) + 1, n - \lambda + 1)$

$\Rightarrow$ $p_{a,i}$: the item a is at the location i in the candidate pattern p

- The first symbol must be in Σ:

$$\bigvee_{a \in \Sigma} p_{a,0} \tag{1}$$

An Encoding of Frequent Pattern Mining

- The locations where the candidate pattern does not appear:

$$\bigwedge_{a\in\Sigma,0\leqslant l\leqslant n-1,0\leqslant i\leqslant k_a-1} (p_{a,i} \wedge s_{l+i} \neq a) \rightarrow b_l \tag{2}$$

- Frequency Constraint:

$$\sum_{l=0}^{n-1} b_l \leqslant n - \lambda \tag{3}$$

Example

Sequence ($\lambda = 2$):

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

Boolean formula:

$$\begin{aligned}
&p_{a,0} \vee p_{b,0} \\
&p_{a,0} \rightarrow (b_2 \wedge b_3) \\
&p_{a,1} \rightarrow (b_1 \wedge b_2 \wedge b_3) \\
&p_{a,2} \rightarrow (b_0 \wedge b_1 \wedge b_2 \wedge b_3) \\
&p_{b,0} \rightarrow (b_0 \wedge b_1) \\
&p_{b,1} \rightarrow (b_0 \wedge b_3) \\
&p_{b,2} \rightarrow (b_2 \wedge b_3) \\
&b_0 + b_1 + b_2 + b_3 \leqslant 2
\end{aligned}$$

Models: $\{p_{a,0}\} \leftrightarrow a$, $\{p_{b,0}\} \leftrightarrow b$ and $\{p_{a,0}, p_{b,2}\} \leftrightarrow a \circ b$.

Closed Patterns

- All the locations where the candidate pattern appears (iff with (2)): $b_l = \mathit{false}$ iff $P \preceq_l S$

$$\bigwedge_{l=0}^{n-1} (b_l \rightarrow \bigvee_{a \in \Sigma, 0 \leqslant i \leqslant k_a - 1} (p_{a,i} \wedge s_{l+i} \neq a)) \tag{4}$$

- Maximizing the number of symbols different from wildcard on the right side:

$$\bigwedge_{a \in \Sigma, 0 \leqslant i \leqslant k_a - 1} (\bigwedge_{l=0}^{n-1} \overline{b_l} \rightarrow s_{l+i} = a) \rightarrow p_{a,i} \tag{5}$$

Closed Patterns

- Maximizing the number of symbols different from wildcard on the left side (negative indices):

$$\bigwedge_{a\in\Sigma,1\leqslant i\leqslant k'_a} (\bigwedge_{l=0}^{n-1} \overline{b_l} \rightarrow s_{l-i} = a) \leftrightarrow p_{a,-i} \tag{6}$$

where $k'_a = n - min(L_S(a)) - 1$

$\Rightarrow$ Without negative indices:

$$\bigwedge_{a\in\Sigma,1\leqslant i\leqslant k'_a} \neg(\bigwedge_{l=0}^{n-1} \overline{b_l} \rightarrow s_{l-i} = a) \tag{7}$$

- Frequent Closed Patterns: (1), (2), (3), (4), (5) and (7)

Maximal Patterns

- Maximizing the number of symbols different from wildcard on the right side:

$$\bigwedge_{a\in\Sigma,1\leqslant i\leqslant k_a-1}(\sum_{l=0}^{n-1}\overline{b_l}\wedge s_{l+i}=a\geqslant\lambda)\rightarrow p_{a,i} \tag{8}$$

- Maximizing the number of symbols different from wildcard on the left side:

$$\bigwedge_{a\in\Sigma,1\leqslant i\leqslant k'_a}\sum_{l=0}^{n-1}\overline{b_l}\wedge s_{l-i}=a\leqslant\lambda-1 \tag{9}$$

- Frequent Maximal Patterns: (1), (2), (3), (4), (8) and (9)

Flexibility of the proposed approach

- The frequent patterns with at least *min* items:

$$\sum_{a \in \Sigma, 0 \leqslant i \leqslant k_a - 1} p_{a,i} \geqslant min \tag{10}$$

- The frequent patterns with at most *max* items:

$$\sum_{a \in \Sigma, 0 \leqslant i \leqslant k_a - 1} p_{a,i} \leqslant max \tag{11}$$

Outline

Sequences of Itemsets

- Sequences of itemsets S: $s_0 \dots s_{n-1}$ s.t. $s_i \subseteq \Sigma$
- Wildcard: $\emptyset$
- Inclusion: $P_0 P_1 \dots P_{m-1} \preceq_I P'_0 P'_1 \dots P'_{n-1}$ if,
  $\forall i \in \{0, \dots, m-1\}$, $P_i \subseteq P'_{I+i}$
- Examples:
  $S = \{a, b\}, \{a, b\}, \{c, d\}, \{c, e\}, \{f\}, \{g\}, \{d\}, \{a, b, d\}, \{f\}, \{c\}$
  $P = \{a, b\}, \{\}, \{c\}$
  $L_S(P) = \{0, 1, 7\}$
- Frequent (Closed/Maximal) patterns are defined in the same way as in the case of the sequences of items

Frequent Pattern Mining

- The Boolean variable $p_{a,i}$ means that the symbol a is in the itemset at the location i in the candidate pattern
- We only have to replace $s_{l+i} \neq a$ with $a \notin s_{l+i}$:

$$\bigvee_{a \in \Sigma} p_{a,0} \tag{12}$$

$$\bigwedge_{a \in \Sigma, 0 \leqslant l \leqslant n-1, 0 \leqslant i \leqslant k_a - 1} (p_{a,i} \wedge a \notin s_{l+i}) \rightarrow b_l \tag{13}$$

$$\sum_{l=0}^{n-1} b_l \leqslant n - \lambda \tag{14}$$

Closed Patterns

Constraints of closeness:

$$\bigwedge_{l=0}^{n-1}(b_l \rightarrow \bigvee_{a\in\Sigma,0\leqslant i\leqslant k_a-1}(p_{a,i} \wedge a \notin s_{l+i})) \tag{15}$$

$$\bigwedge_{a\in\Sigma,0\leqslant i\leqslant k_a-1}(\bigwedge_{l=0}^{n-1}\overline{b_l} \rightarrow a \in s_{l+i}) \rightarrow p_{a,i} \tag{16}$$

$$\bigwedge_{a\in\Sigma,1\leqslant i\leqslant k'_a}\neg(\bigwedge_{l=0}^{n-1}\overline{b_l} \rightarrow a \in s_{l-i}) \tag{17}$$

Maximal Patterns

Constraints of maximality:
we add the following two constraints to (15):

$$\bigwedge_{a\in\Sigma,1\leqslant i\leqslant k_a-1} (\sum_{l=0}^{n-1} \overline{b_l} \wedge a \in s_{l+i} \geqslant \lambda) \rightarrow p_{a,i} \tag{18}$$

$$\bigwedge_{a\in\Sigma,1\leqslant i\leqslant k'_a} \sum_{l=0}^{n-1} \overline{b_l} \wedge a \in s_{l-i} \leqslant \lambda - 1 \tag{19}$$

Outline

Symmetries

- A fundamental concept (structural knowledge) in Computer Science, Mathematics, Physics and many other domains.
  - Many human artifacts (e.g. classrooms 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 complex entities and systems.

Frequent Itemset Mining

- Essential problem in data mining, knowledge discovery and data analysis.
- Many related problems: Association rules, frequent pattern mining in sequence data, data clustering, episode mining, etc.
- Various applications

Frequent Itemset Mining

- Essential problem in data mining, knowledge discovery and data analysis.
- Many related problems: Association rules, frequent pattern mining in sequence data, data clustering, episode mining, etc.
- Various applications

Main challenges

- Output of huge size, difficulty to retrieve relevant information
- Computational issues

Frequent Itemset Mining: Problem definition and notations

- Let $\mathcal{I}$ be a set of *items*.
- A set $I \subseteq \mathcal{I}$ is called an **itemset**.
- A **transaction** is a couple (t_i, I) where t_i is the *transaction identifier* and I is an itemset.
- A **transaction database** is a finite set of transactions over $\mathcal{I}$ where for each two different transactions, they do not have the same transaction identifier.
- **Cover**: $\mathcal{C}(I, \mathcal{D}) = \{t_i \mid (t_i, J) \in \mathcal{D} \text{ and } I \subseteq J\}$.
- **Support**: $\mathcal{S}(I, \mathcal{D}) = |\mathcal{C}(I, \mathcal{D})|$.
- **Frequency**: $\mathcal{F}(I, \mathcal{D}) = \frac{\mathcal{S}(I,\mathcal{D})}{|\mathcal{D}|}$.

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

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

$\mathcal{I} = \{A, B, C, D, E, F\}$

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

- $\mathcal{I} = \{A, B, C, D, E, F\}$
- **Itemset:** $I \subseteq \mathcal{I}$.

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

- $\mathcal{I} = \{A, B, C, D, E, F\}$
- **Itemset:** $I \subseteq \mathcal{I}$.
- **Cover:** $\mathcal{C}(\{A, C\}, \mathcal{D}) = \{002, 004\}$

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

- $\mathcal{I}$= $\{A, B, C, D, E, F\}$
- **Itemset:** $I \subseteq \mathcal{I}$.
- **Cover:** $\mathcal{C}(\{A, C\}, \mathcal{D})$= $\{002, 004\}$
- **Support:** $\mathcal{S}(\{A, C\}, \mathcal{D})$= $|\{002, 004\}|$=2

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

- $\mathcal{I}$= $\{A, B, C, D, E, F\}$
- **Itemset:** $I \subseteq \mathcal{I}$.
- **Cover:** $\mathcal{C}(\{A, C\}, \mathcal{D})$= $\{002, 004\}$
- **Support:** $\mathcal{S}(\{A, C\}, \mathcal{D})$= $|\{002, 004\}|$=2
- **Frequency:** $\mathcal{F}(\{A, C\}, \mathcal{D})$=$2/9 \equiv 0.22$

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

- $\mathcal{I}$= $\{A, B, C, D, E, F\}$
- **Itemset:** $I \subseteq \mathcal{I}$.
- **Cover:** $\mathcal{C}(\{A, C\}, \mathcal{D})$= $\{002, 004\}$
- **Support:** $\mathcal{S}(\{A, C\}, \mathcal{D})$= $|\{002, 004\}|$=2
- **Frequency:** $\mathcal{F}(\{A, C\}, \mathcal{D})$=$2/9 \equiv 0.22$

Definition (Frequent Itemset Mining Problem)

Given a minimum support λ ($0 < \lambda \leqslant |\mathcal{D}|$), the frequent itemset mining problem consists in computing the set of itemsets

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

Definition (Frequent Itemset Mining Problem)

Given a minimum support λ ($0 < \lambda \leqslant |\mathcal{D}|$), the frequent itemset mining problem consists in computing the set of itemsets

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

Proposition (Anti-Monotonicity)

Let I_1 and I_2 be two itemsets such that $I_1 \subseteq I_2$.

$$\textit{If } \mathcal{S}(I_2, \mathcal{D}) \geqslant \lambda \textit{ then } \mathcal{S}(I_1, \mathcal{D}) \geqslant \lambda.$$

Outline

Definition (Permutation)

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

Definition (Permutation)

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

Definition (Symmetry)

A permutation σ over $\mathcal{I}$ is a symmetry if $\sigma(\mathcal{D}) = \mathcal{D}$ where $\sigma(\mathcal{D}) = \{\sigma(t_i, l) = (\sigma(t_i), \sigma(l)), (t_i, l) \in \mathcal{D}\}$

Definition (Permutation)

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

Definition (Symmetry)

A permutation σ over $\mathcal{I}$ is a symmetry if $\sigma(\mathcal{D}) = \mathcal{D}$ where $\sigma(\mathcal{D}) = \{\sigma(t_i, l) = (\sigma(t_i), \sigma(l)), (t_i, l) \in \mathcal{D}\}$

$\sigma = c_1 \ldots c_n$ 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 (Permutation)

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

Definition (Symmetry)

A permutation σ over $\mathcal{I}$ is a symmetry if $\sigma(\mathcal{D}) = \mathcal{D}$ where $\sigma(\mathcal{D}) = \{\sigma(t_i, I) = (\sigma(t_i), \sigma(I)), (t_i, I) \in \mathcal{D}\}$

$\sigma = c_1 \ldots c_n$ 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 $\sigma(a_k) = a_1$.

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 $\sigma(I) \in \mathcal{FIM}(\mathcal{D}, \lambda)$.

Example

σ = (C,E)(D,F) is a symmetry

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

Example

σ = (C,E)(D,F) is a symmetry

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

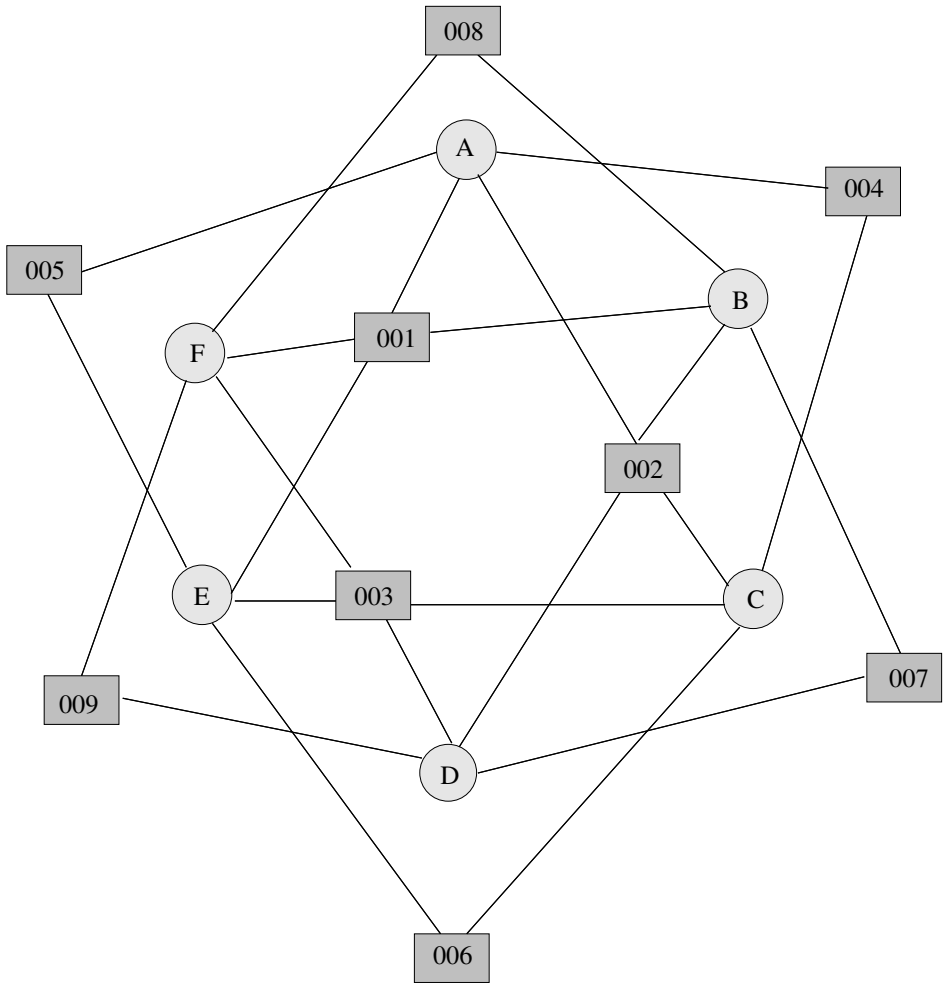
$$\sigma(t_i) = \begin{cases} 001 & \text{if } t_i\text{=002} \\ 002 & \text{if } t_i\text{=001} \\ 003 & \text{if } t_i\text{=003} \\ 004 & \text{if } t_i\text{=005} \\ 005 & \text{if } t_i\text{=004} \\ 006 & \text{if } t_i\text{=006} \\ 007 & \text{if } t_i\text{=008} \\ 008 & \text{if } t_i\text{=007}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

Outline

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 $\mathcal{G}$ ([Jabbour et al, ECAI'12]);
- Employ a general-purpose graph symmetry tool to uncover the symmetries [Mckay'81, Aloul'03].

Symmetry Detection in Transaction Databases: 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		

Outline

How to exploit symmetries in itemset mining?

1. By rewriting the transaction databases in a preprocessing step (items elimination). [Jabbour et al, ECAI'12]
   - $\rightarrow$ New transaction database D' + symmetry group S.
   - $\rightarrow$ Condensed representation of the output.
2. By dynamic integration in Apriori-like algorithms for search space pruning.

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.

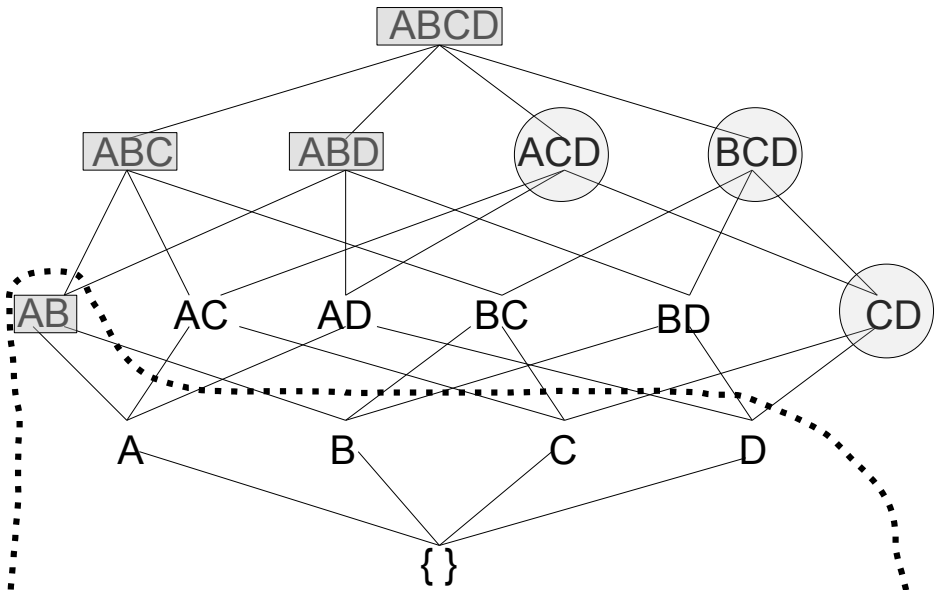


Figure : Symmetry Pruning

Symmetry Breaking

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

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

$$\begin{array}{lcl} \{a, \dots\} & \rightarrow & \{b, \dots\} \\ \{a, d, \dots\} & \rightarrow & \{b, c, \dots\} \\ \{a, c, \dots\} & \rightarrow & \{b, d, \dots\} \\ \{a, b, \dots\} & \rightarrow & \{a, b, \dots\} \end{array} \qquad \begin{array}{lcl} \{b, \dots\} & \rightarrow & \{a, \dots\} \\ \{b, c, \dots\} & \rightarrow & \{a, d, \dots\} \\ \{d, \dots\} & \rightarrow & \{c, \dots\} \\ \{b, d, \dots\} & \rightarrow & \{a, c, \dots\} \end{array}$$

Symmetry Breaking

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

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

$$
\begin{array}{lcl}
\{a, \dots\} & \rightarrow & \{b, \dots\} \\
\{a, d, \dots\} & \rightarrow & \{b, c, \dots\} \\
\{a, c, \dots\} & \rightarrow & \{b, d, \dots\} \\
\{a, b, \dots\} & \rightarrow & \{a, b, \dots\}
\end{array}
\qquad
\begin{array}{lcl}
\{b, \dots\} & \rightarrow & \{a, \dots\} \\
\{b, c, \dots\} & \rightarrow & \{a, d, \dots\} \\
\{d, \dots\} & \rightarrow & \{c, \dots\} \\
\{b, d, \dots\} & \rightarrow & \{a, c, \dots\}
\end{array}
$$

Symmetry Breaking

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

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

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

- $\Rightarrow$ 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

$\sigma = (x_1, y_1)(x_2, y_2) \dots (x_j, y_j) \dots (x_n, y_n)$ 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

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 : Symmetry Breaking approach

Outline

Propositional logic

- Classical propositional logic:

$$A ::= p \mid \neg A \mid A \wedge A \mid A \vee A \mid A \to A$$

- De Morgan laws:

$$A \vee B \equiv \neg(\neg A \wedge \neg B) \quad A \wedge B \equiv \neg(\neg A \vee \neg B)$$
$$A \to B \equiv \neg A \vee B \qquad \neg\neg A \equiv A$$

- Boolean interpretation:
  - $[\![\cdot]\!] : \text{Prop} \to \{0, 1\}$
  - Extension to formulae: $[\![\neg A]\!] = 1 - [\![A]\!]$, $[\![A \wedge B]\!] = min([\![A]\!], [\![B]\!])$

- Satisfiability: $\exists [\![\cdot]\!], [\![A]\!] = 1$ (NP-complete [Cook 71])

Conjunctive Normal Form (CNF) and SAT

- A conjunction of clauses:

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

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

- Example :

$$(p \vee \neg q \vee \neg r) \wedge (p \vee \neg q \vee s) \wedge p \wedge (r \vee \neg s)$$

Conjunctive Normal Form (CNF) and SAT

- A conjunction of clauses:

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

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

- Example :

$$\overbrace{(p \vee \neg q \vee \neg r)}^{1} \wedge \overbrace{(p \vee \neg q \vee s)}^{1} \wedge \overbrace{p}^{1} \wedge (r \vee \neg s)$$

$$[\![p]\!] = 1$$

Conjunctive Normal Form (CNF) and SAT

- A conjunction of clauses:

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

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

- Example :

$$\overbrace{(p \vee \neg q \vee \neg r)}^{1} \wedge \overbrace{(p \vee \neg q \vee s)}^{1} \wedge \overbrace{p}^{1} \wedge \overbrace{(r \vee \neg s)}^{1}$$

$[\![p]\!] = 1$ et $[\![r]\!] = 1$ (Partial interpretation)

Transformation - Extension principle [G. Tseitin 1965]

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

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

- Nave 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
  $(z_1 \vee \cdots \vee z_n) \wedge (\neg z_1 \vee x_1) \wedge (\neg z_1 \vee y_1) \wedge \cdots \wedge (\neg z_n \vee x_n) \wedge$
  $(\neg z_n \vee y_n)$

Extended resolution proof system [G. Tseitin 1965]

- Extended resolution:

$$\frac{l \vee \alpha \in \Sigma \qquad \bar{l} \vee \beta \in \Sigma}{\alpha \vee \beta} \; [Res]$$

$$\text{Extension : } x \leftrightarrow F$$

- Shorten resolution proofs

- Open question: automatization of extended resolution proof systems ?

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:
    $\bigwedge_{l=1}^{n} \bigwedge_{c=1}^{n} (\bigvee_{v=1}^{n} p_{(l,c,v)}) \quad \Longrightarrow \quad n^2$ 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 $x_3 = x_1 \wedge x_2$
1 -2 3 0
. . . 1million pages later
-11482897 -11483041 -11483523 0
11482897 11483041 -11483523 0 $x_3 \leftrightarrow x_4 \leftrightarrow x_5$
11482897 -11483041 11483523 0
-11482897 11483041 11483523 0
-11483518 -11483524 0
-11483519 -11483524 0
-11483520 -11483524 0
-11483521 -11483524 0 $x_6 = (x_7 \wedge x_8 \wedge x_9 \wedge x_{10} \wedge x_{11} \wedge x_{12})$
-11483522 -11483524 0
-11483523 -11483524 0
11483518 11483519 11483520 11483521 11483522 11483523 11483524 0
-8590303 -11483524 -11483525 0
8590303 11483524 -11483525 0 $x_{13} \leftrightarrow x_{14} \leftrightarrow x_{15}$
8590303 -11483524 11483525 0
-8590303 11483524 11483525 0
-11483525 0

Frequent itemsets mining

- Transactions database

tid	itemset
001	*Joyce*, *Beckett*, *Proust*
002	*Faulkner*, *Hemingway*, *Melville*
003	*Joyce*, *Proust*
004	*Hemingway*, *Melville*
005	*Flaubert*, *Zola*
006	*Hemingway*, *Golding*

- Support: $\mathcal{S}(\{\textit{Hemingway}, \textit{Melville}\}, \mathcal{D}) = |\{002, 004\}| = 2$
- Enumerating frequent itemsets:
  $\mathcal{FIM}(\mathcal{D}, \lambda) = \{A \subseteq \mathcal{I} | \mathcal{S}(A, \mathcal{D}) \geqslant \lambda\}$
- Example: $\mathcal{FIM}(\mathcal{D}, 2) =$
  $\{\{\textit{Hemingway}\}, \{\textit{Melville}\}, \{\textit{Hemingway}, \textit{Melville}\}, \{\textit{Joyce}\},$
  $\{\textit{Proust}\}, \{\textit{Joyce}, \textit{Proust}\}\}$

Frequent itemsets mining

Condensed representations of frequent itemsets

- Maximal frequent itemsets:

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

- Closed frequent itemsets:

$$Cl(\mathcal{D}, \lambda) = \{A \in \mathcal{FIM}(\mathcal{D}, \lambda) | \forall B \supset A, \mathcal{S}(B, \mathcal{D}) \neq \mathcal{S}(A, \mathcal{D})\}$$

- Example :

$$Max(\mathcal{D}, 2) = \{\{Joyce, Proust\}, \{Hemingway, Melville\}\}$$

$$Cl(\mathcal{D}, 2) = \{\{Hemingway\}, \{Joyce, Proust\}, \{Hemingway, Melville$$

CNF formula as transactions database

- Goal : reduce the number of literals using the frequent sets of literals: similar to Tseitin approach (introduce new Boolean variables)
- Items: literals
- Transactions: clauses > 2

Reduce the number of literals

- Introduce new Boolean variables:

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

equivalent w.r.t. *SAT*

$$\Longrightarrow$$

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

- $n \geqslant 2$ et $k > \frac{n+1}{n-1}$

- $n \times k$ literals substituted by $k + n + 1$ literals
- Quorum : $k \begin{cases} \geqslant 4 & \text{si } n = 2 \\ \geqslant 3 & \text{si } n = 3 \\ \geqslant 2 & \text{otherwise} \end{cases}$
- Not interesting to associate new variables to subsets of $\{x_1, \ldots, x_n\}$: use of condensed representation

Closed Vs. Maximal

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

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

with $k \geqslant 2$ and $m, m' \geqslant 4$
we suppose that the set of itemsets are frequent
$\mathcal{P}(\{x_1, \ldots, x_n\})$
$\Rightarrow$ Max $= \{\{x_1, \ldots, x_n\}\}$ and closed
$= \{\{x_1, \ldots, x_k\}, \{x_1, \ldots, x_n\}\}$

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

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

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

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

Subsets

- X and Y ($Y \subset X$) are both interesting if

$$\mathcal{S}(Y) - \mathcal{S}(X) > \frac{|Y|+1}{|Y|-1} - 1$$

- 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

- X overlaps with Y ($X \sim Y$): $X \cap Y \neq \emptyset$
- overlaps class (Overlap class) : an equivalence class (transitive closure of $\sim$)

$$Y \in [X] \;\; ssi \;\; Y = Y_1 \sim Y_2 \sim \ldots \sim Y_k = X$$

- Optimal solution $\longrightarrow$ optimal solution in an overlaps class

Overlap

- Problem :
  - $\{x_1, x_2, x_3\}$ 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 $\{x_1, x_2, x_3\} \longrightarrow \mathcal{S}(\{x_2, x_3, x_4\}) = 2$

- X and Y ($Y \sim X$) are both interesting if
  - $\mathcal{S}(X) - \mathcal{S}(X \cup Y) > \frac{|X|+1}{|X|-1} - 1$,
  - $\mathcal{S}(Y) - \mathcal{S}(X \cup Y) > \frac{|Y|+1}{|Y|-1} - 1$, or
  - $|X \setminus Y| \geqslant k$ (resp. $|Y \setminus X| \geqslant k$) where
    - $k = 2$ if $\mathcal{S}(X) \geqslant 4$ (resp. $\mathcal{S}(Y) \geqslant 4$)
    - $k = 3$ if $\mathcal{S}(X) = 3$ (resp. $\mathcal{S}(Y) = 3$)
    - $k = 4$ otherwise

    $\Rightarrow$ Use of $X \setminus Y$ (resp. $Y \setminus X$)

Problems summary

- Choose of the quorum ?
  - 2 $\longrightarrow$ lot of useless itemsets
  - 3 and 4 $\longrightarrow$ loss of interesting itemsets
- Overlap (subsets) :
  - Simplification: overlaps classes
  - Need to compute $\mathcal{S}(X \cup Y)$
  - Optimal solution in one class?
- A gready algorithm: loss of interesting itemsets

Gready Algorithm

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

1: **while** $C \neq \emptyset$ **do**
2: $I \leftarrow C.MostInterstingElement()$;
3: $\phi.replace(I, x)$;
4: $\phi.Add(I, x)$:
5: $C.remove(I)$;
6: $C.replaceSubset(I, x)$;
7: $C.removeUninterestingElements()$;
8: $C.updateSupports()$;
9: **end while**
10: **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

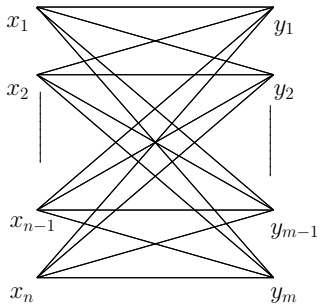
FIM process on the conjunctive part of $B^1_{\vee[\wedge]}(\Phi)$

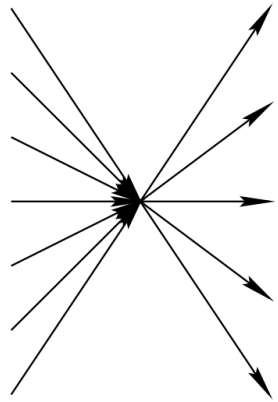
Using $\{x_5, x_6, x_7\}$ a 4-frequent itemset, we can rewrite $B^1_{[\vee(\wedge)]}(\Phi)$ as:

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

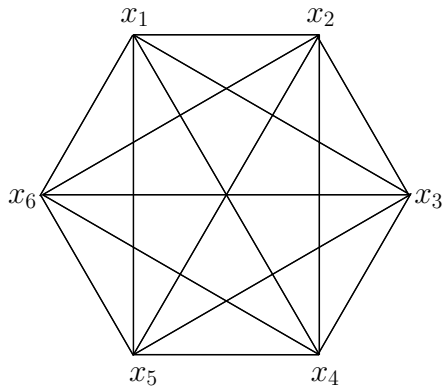
$$
\begin{aligned}
\mathrm{CNF}(B^2_{[\vee(\wedge)]}(\Phi)) = \\
& (x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_1 \vee y) && \wedge \\
& (x_2 \vee x_3) \wedge (x_2 \vee x_4) \wedge (x_2 \vee y) && \wedge \\
& (x_3 \vee x_4) \wedge (x_3 \vee y) && \wedge \\
& (x_5 \vee x_6) \wedge (x_5 \vee x_7) && \wedge \\
& (x_6 \vee x_7) && \wedge \\
& (\neg y \vee x_5) \wedge (\neg y \vee x_6) \wedge (\neg y \vee x_7)
\end{aligned}
$$

Two particular cases: bi-cliques and cliques





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



$\mathcal{O}(n^2)$ binary clauses $\Rightarrow$ $\mathcal{O}(n)$ binary clauses and $\mathcal{O}(n)$ new variables

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

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
- $2 \rightarrow [4, 6, 8, 12] \leftrightarrow 2 \vee [4 \wedge 6 \wedge 8 \wedge 12]$

Compression as an Optimisation Problem

The compression problem can be formulated as an optimisation problem

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

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

Compression as an Optimisation Problem

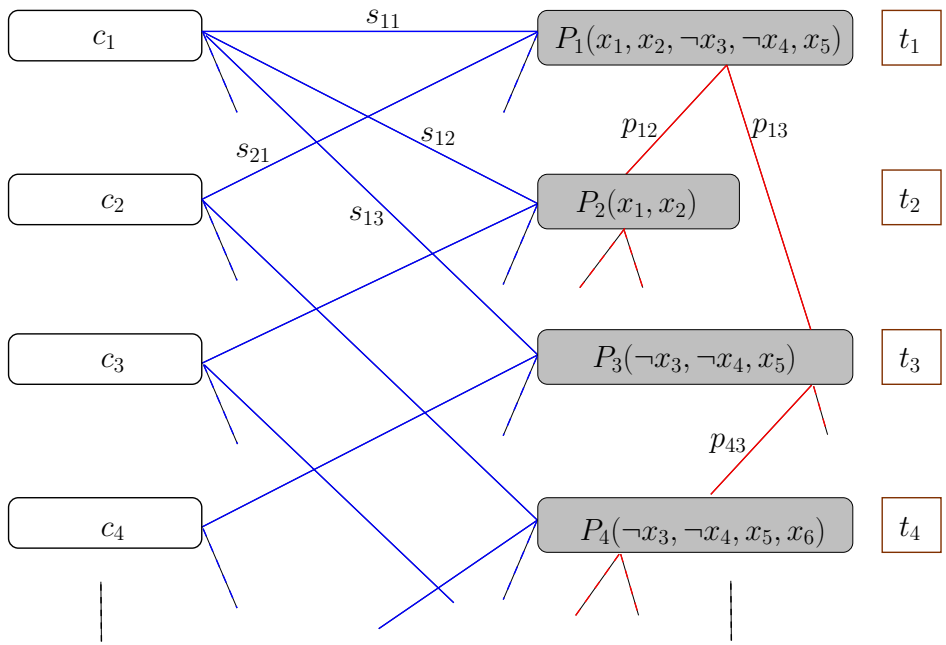
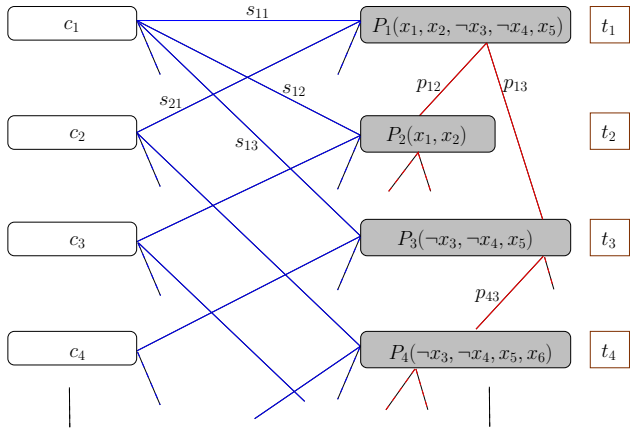


Figure : Compression using location problem

- If we use pattern P_j, we set t_j to 1, otherwise t_j is 0
- If we replace the literals in c_i by P_i, then we set s_{ij} to 1, and 0 otherwise.

Compression as an Optimisation Problem

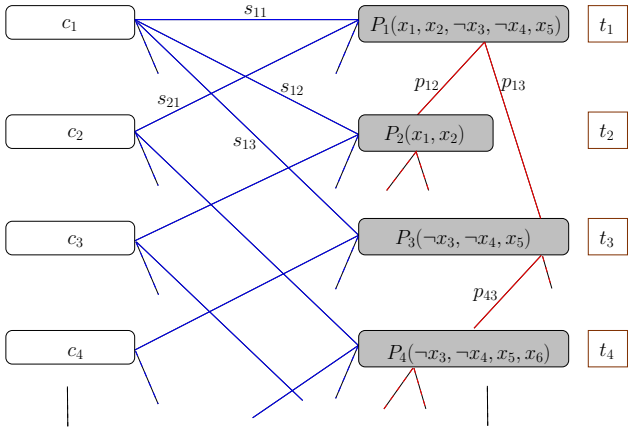


A **first formulation** as 0/1 linear program

$Max \sum(|P_j| - 1)s_{ij} - \sum(|P_j| + 1)t_j$ Maximize the reduction

1. $s_{ij} \leqslant t_j$ (C_i is replaced by P_j only when P_j is used)
2. $\sum_j s_{ij} \leqslant 1$ (C_i is replaced by only one pattern)
3. $s_{ij} \in \{0, 1\}$, $t_j \in \{0, 1\}$

Compression as an Optimisation Problem



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

$Max \sum(|P_j| - 1)s_{ij} - \sum(|P_j| + 1)t_j$ Maximize the reduction

1. $s_{ij} \leqslant t_j$ (C_i is replaced by P_j only when P_j is used)
2. $s_{ij} + s_{ik} \leqslant 1$ if $P_j \cap P_k \neq \emptyset$ (C_i can be replaced by a set of disjoint patterns)
3. $s_{ij} \in \{0, 1\}$, $t_j \in \{0, 1\}$

Outline

Conclusions

- Theoretical foundations for discovering and using symmetries in itemset mining problems.
- Using symmetries to prune a search space.
- Integration of symmetry-based pruning in Apriori-like algorithms.

Conclusions

- Theoretical foundations for discovering and using symmetries in itemset mining problems.
- Using symmetries to prune a search space.
- Integration of symmetry-based pruning in Apriori-like algorithms.

Futur works

- Extend the symmetry-based framework to other data mining algorithms and problems : sequence, tree or graph mining, etc.
- Investigate other forms of symmetries such as approximate symmetries.