SAT-Based Data Mining

Saïd Jabbour

CRIL - CNRS UMR 8188
Université d'Artois, France

GDR-IA - GT CAVIAR

Orléans

May 27, 2019

Outline

Frequent Itemsets Mining

Propositional Logic and SAT problem

(Parallel) SAT-based Solvers for Enumerating all (C, M)FIM on on (Uncertain) Transaction Databases

Association Rules Mining

Gradual Itemsets Mining

Symmetry Breaking in Frequent Itemsets Mining

FIM for CNF Formulas compression

Data Mining

- Discovering interesting knowledge from large amounts of data.
  - Frequent itemsets
  - Sequential patterns
  - Association rules
  - Emerging patterns
  - ...
- Frequent itemset mining is an important part of data mining.
- Different variety of applications : **Healthcare, Business, Education, Disaster prevention, etc.**

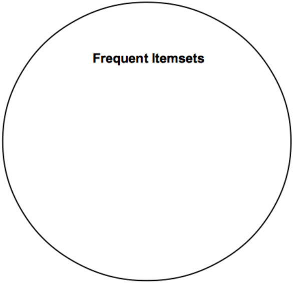
Frequent Itemset Mining

- A set of **items :** $\Omega = \{a, b, c, \ldots\}$.
- An **itemset** I over Ω : is a subset of Ω, i.e., $I \subseteq \Omega$.
- **A transaction :** couple (tid, I)
  tid is the *transaction identifier* and
  I is an *itemset*, i.e., $I \subseteq \Omega$.
- **Transaction database $\mathcal{D}$:** set of transactions.

TID	Transactions				
T_1	a	b	c	d	
T_2	a	b	c		e
T_3	a				e
T_4	a			d	e
T_5	a	b			
T_6		b		d	
T_7		b			e

- A transaction (tid, I) **supports** an itemset J if $J \subseteq I$.
- The **cover** of an itemset I :
  $\mathbf{Cover(I, \mathcal{D}) = \{tid \mid (tid, J) \in \mathcal{D}, I \subseteq J\}}$.
  - $Cover(\{ab\}, \mathcal{D}) = \{T_1, T_2, T_5\}$
- The **support** of an itemset I in $\mathcal{D}$: $\mathbf{Supp(I, \mathcal{D}) = \mid Cover(I, \mathcal{D}) \mid}$.
  - $Supp(\{ab\}, \mathcal{D}) = 3$

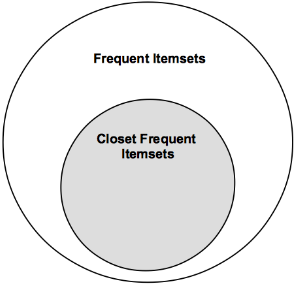
Frequent Itemset Mining



- **FIM**$(\mathcal{D}, \theta) = \{I \subseteq \Omega \mid Supp(I, \mathcal{D}) \geq \theta\}$

- An itemset I is frequent if its support is greater than or equal to a minsup threshold.

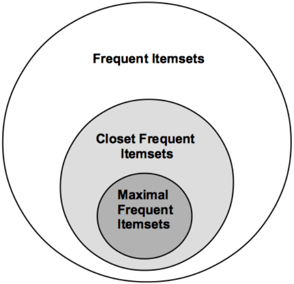
Frequent Itemset Mining



- **FIM**$(\mathcal{D}, \theta) = \{I \subseteq \Omega \mid Supp(I, \mathcal{D}) \geq \theta\}$
- **CFIM**$(\mathcal{D}, \theta) = \{I \in FIM(\mathcal{D}, \theta) \mid \forall J \supset I, Supp(I, \mathcal{D}) > \mathcal{S}(J, \mathcal{D})\}$

- An itemset *I* is closed if *I* is frequent and there exists no super-pattern $J \supset I$, with the same support as *I*.

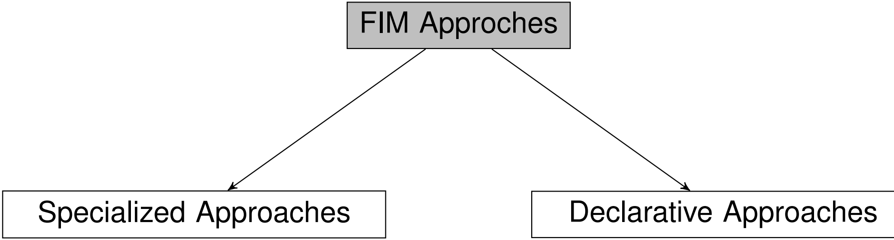
Frequent Itemset Mining



- **FIM**$(\mathcal{D}, \theta) = \{I \subseteq \Omega \mid Supp(I, \mathcal{D}) \geq \theta\}$
- **CFIM**$(\mathcal{D}, \theta) = \{I \in FIM(\mathcal{D}, \theta) \mid \forall J \supset I, Supp(I, \mathcal{D}) > \mathcal{S}(J, \mathcal{D})\}$
- **MFIM**$(\mathcal{D}, \theta) = \{I \in FIM(\mathcal{D}, \theta) \mid \forall J \supset I, Supp(J, \mathcal{D}) < \theta\}$

An itemset I is a max-pattern if I is frequent and there exists no frequent super-pattern $J \supset I$.

Frequent Itemset Mining



- **Apriori** [Agrawal'93]
- **FP-growth** [Han'00]
- **ECLAT** [Zaki'00]
- **LCM** [Un'04], ...

- **CP** [De Raedt'08]
- **SAT** [Jabbour'13]
- **ASP** [Gebser'16]
- **...**

Propositional Logic

Formal Language of propositional formulas : $\mathcal{P}rop$

Syntax

- Logical constant : $\bot$, $\top$
- Propositional symbols : a, b, c, ... (atomic sentences)
- Wrapping parentheses : (...)
- Sentences are combined by connectives : $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\Leftrightarrow$.

If $\Phi_1, \ \Phi_2 \in \mathcal{P}rop$, then the following formulas are in $\mathcal{P}rop$:

$$\begin{array}{lll} \neg\Phi_1 & (\Phi_1 \wedge \Phi_2) & (\Phi_1 \vee \Phi_2) \\ & (\Phi_1 \rightarrow \Phi_2) & (\Phi_1 \Leftrightarrow \Phi_2) \end{array}$$

Propositional Logic : SAT

Semantic : an interpretation is a fonction from $\mathcal{P}rop$ to $\{0,1\}$ (0 : false ; 1 : true).
Defined inductively as :

$$\mathcal{B} : \left[\begin{array}{lll} \mathcal{P}rop & \rightarrow & \{0,1\} \\ & & \\ \bot & & 0 \\ \top & & 1 \\ F \wedge G & & min(\mathcal{B}(F), \mathcal{B}(G)) \\ \neg F & & 1 - \mathcal{B}(F) \\ F \vee G & & max(\mathcal{B}(F), \mathcal{B}(G)) \end{array} \right.$$

- A **model** of Φ is an interpretation $\mathcal{B}$ satisfying Φ, i.e., $\mathcal{B}(\Phi) = 1$.
- A formula Φ is **satisfiable** if there exists a model of Φ.

Propositional logic : SAT

SAT problem : decide if a formula in CNF is satisfiable or not ? [NP-Complete'71]

CNF :	conjunction of clauses	$c_1 \wedge \ldots \wedge c_n$
Clause :	disjunction of literals	$(l_1 \ldots \vee l_k)$
Literal :	a variable or its negation	$\{l_i,\ \neg l_i\}$

$$\Phi = \overbrace{(a \vee b \vee c)}^{c_1} \wedge \overbrace{(\neg a \vee b)}^{c_2} \wedge \overbrace{(b \vee c)}^{c_3} \wedge \overbrace{(\neg c \vee a)}^{c_4}$$

Various Applications : Model Checking, Planning, Data Mining, etc.

$\rightarrow$ easier formulation

$\rightarrow$ efficient solving

SAT Problem

- **Models enumeration problem**
  - Variant of the propositional satisfiability problem (SAT)

$$\Phi = \overbrace{(a \vee b \vee c)}^{C_1} \wedge \overbrace{(\neg a \vee b)}^{C_2} \wedge \overbrace{(b \vee c)}^{C_3} \wedge \overbrace{(\neg c \vee a)}^{C_4}$$

$$\mathcal{M}(\Phi) = \left\{ \begin{array}{ll} \{a = 1, b = 1, c = 1\} & \{a = 0, b = 1, c = 0\} \\ \{a = 1, b = 1, c = 0\} & \{a = 0, b = 1, c = 0\} \end{array} \right\}$$

  - Different application domains :
    - Data mining
    - Bounded model checking
    - Knowledge compilation
    - ...
- Models enumeration problem received little attention compared to other SAT issues.

Itemsets Mining

Ω	items (finite set of symbols)
I	Itemset (subset of Ω)
$T_i = (i, I_i)$	Transaction with $i \in \mathbb{N}$ the transaction identifier, I_i an itemset
D	Transactional database (set of transactions)

id	*transactions*						
1			*c*	*d*	*e*	*f*	*g*
2			*c*	*d*	*e*	*f*	*g*
3	*a*	*b*	*c*	*d*			
4	*a*	*b*	*c*	*d*		*f*	
5	*a*	*b*	*c*	*d*			
6			*c*		*e*		

id	*transactions*						
1	0	0	1	1	1	1	1
2	0	0	1	1	1	1	1
3	1	1	1	1	0	0	0
4	1	1	1	1	0	1	0
5	1	1	1	1	0	0	0
6	0	0	1	0	1	0	0
	a	*b*	*c*	*d*	*e*	*f*	*g*

Symbolic approach [ECML/PKDD'13]

$$\text{Find } \{I \subseteq \Omega \mid |Supp(I, D)| \geq \theta\},\ \theta \in \mathbb{N}$$

Make frequent itemsets extraction as the models enumeration of a CNF formula ((anti-)monotonicity)

$$\underbrace{\bigwedge_{i=1}^{m} (\neg q_i \leftrightarrow \bigvee_{a \in \Omega \setminus T_i} p_a)}_{cover:\ \Phi^{cov}} \qquad \underbrace{\sum_{i=1}^{m} q_i \geq \theta}_{frequency:\ \Phi^{freq}} \qquad \underbrace{\bigwedge_{a \in \Omega} (p_a \vee \bigvee_{T_i \in D \mid a \notin T_i} q_i)}_{closeness:\ \Phi^{clos}}$$

$\neg q_1$	$\leftrightarrow$	p_a	p_b	c	d	e	f	g
$\neg q_2$	$\leftrightarrow$	p_a	p_b	c	d	e	f	g
$\neg q_3$	$\leftrightarrow$	a	b	c	d	p_e	p_f	p_g
$\neg q_4$	$\leftrightarrow$	a	b	c	d	p_e	f	p_g
$\neg q_5$	$\leftrightarrow$	a	b	c	d	p_e	p_f	p_g
$\neg q_6$	$\leftrightarrow$	p_a	p_b	c	p_d	e	p_f	p_g

$$\begin{array}{ll}
(q_3 \vee q_4 \vee q_5 \vee p_a) & \wedge \\
(q_3 \vee q_4 \vee q_5 \vee p_b) & \wedge \\
(p_c) & \wedge \\
(q_6 \vee p_d) & \wedge \\
(q_1 \vee q_2 \vee q_6 \vee p_e) & \wedge \\
(q_1 \vee q_2 \vee q_4 \vee p_f) & \wedge \\
(q_1 \vee q_2 \vee p_e) &
\end{array}$$

$$q_1 + q_2 + q_3 + q_4 + q_5 + q_6 \geq \theta$$

Symbolic approach

Declarativity : easy extension to mine particular patterns (add new constraints)

$$\Phi^{cov} = \bigwedge_{i=1}^{m} (\neg q_i \leftrightarrow \bigvee_{a \in \Omega \setminus T_i} p_a) \qquad \sum_{T \in D} (|\Omega| - |T| + 1) \approx |D| \times |\Omega|$$

$$\Phi^{freq} = \sum_{i=1}^{m} q_i \geq \theta \qquad O(mlog^2(min\_supp))$$

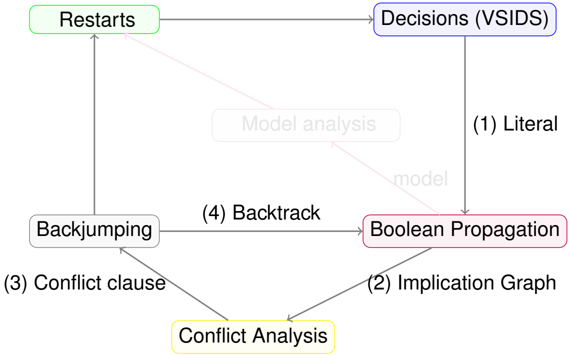
$$\Phi^{clos} = \bigwedge_{a \in \Omega} (p_a \vee \bigvee_{T_i \in D \mid a \notin T_i} q_i) \qquad |D| - |Supp(\{a\})|$$

$$\Phi^{len} = \sum_{a \in \Omega} p_a \geq min\_length$$

Instance	θ	#Tran, #Items	Type of Data	#CFIM
Retail	10	88162, 6470	market basket data	$> 1.10^5$
Kosarak	1000	990002, 41267	hungarian on-line news portal	$\simeq 5.10^5$
accidents	40000	340183, 468	traffic accidents	$\simeq 6.10^6$

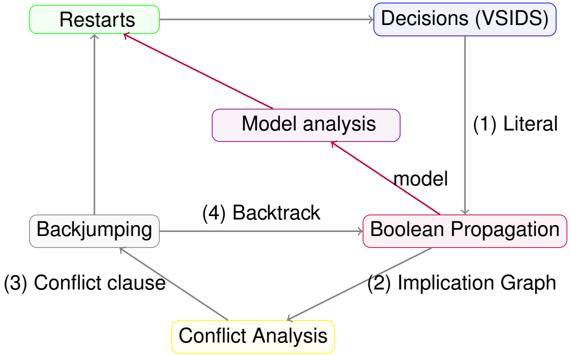
- The number of closed frequent itemsets is often significant.

SAT-based Solvers for Enumerating all CFIM



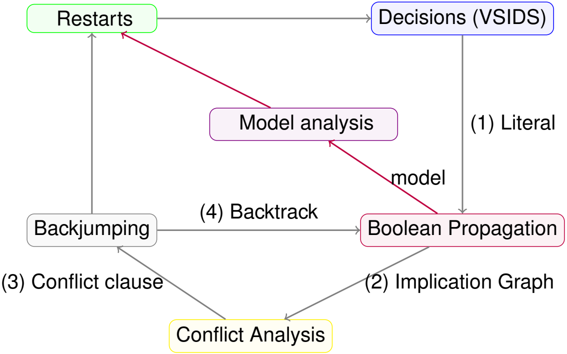
- DPLL SAT-based solver for enumerating CFIM is more efficient

SAT-based Solvers for Enumerating all CFIM



- DPLL SAT-based solver for enumerating CFIM is more efficient

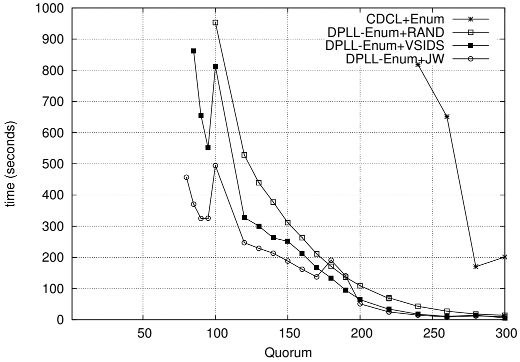
SAT-based Solvers for Enumerating all CFIM



- DPLL SAT-based solver for enumerating CFIM is more efficient

DPLL-based procedure for CFIM [SGAI'16]

- **DPLL-Enum+VSIDS :** Variable State Independent, Decaying Sum branching heuristic
- **DPLL-Enum+JW :** branching heuristic based on the maximum number of occurrences of the variables
- **DPLL-Enum+RAND :** random variable selection



Limitations

$\Phi^{cov} = \bigwedge_{i=1}^{m}(\neg q_i \leftrightarrow \bigvee_{a\in\Omega\setminus T_i} p_a)$	$\sum_{T\in D}(\lvert\Omega\rvert - \lvert T\rvert + 1) \approx \lvert D\rvert \times \lvert\Omega\rvert$
$\Phi^{freq} = \sum_{i=1}^{m} q_i \geq \theta$	$O(mlog^2(min\_supp))$
$\Phi^{clos} = \bigwedge_{a\in\Omega}(p_a \vee \bigvee_{T_i\in D \mid a\notin T_i} q_i)$	$\lvert D\rvert - \lvert Supp(\{a\})\rvert$
$\Phi^{len} = \sum_{a\in\Omega} p_a \geq min\_length$	

Instance	θ	#Tran, #Items	Type of Data	#Clauses	#CFIM
Retail	10	88162, 16470	market basket data	1451119564	$> 1.10^5$
Kosarak	1000	990002, 41267	hungarian on-line news portal	40846393519	$\simeq 5.10^5$
Accidents	40000	340183, 468	traffic accidents	147704774	$\simeq 6.10^6$

- **Scalability problem :** the number of clauses of the SAT encodings is very large.

Limitations

$\Phi^{cov} = \bigwedge_{i=1}^{m} (\neg q_i \leftrightarrow \bigvee_{a \in \Omega \setminus T_i} p_a)$	$\sum_{T \in D} (\lvert\Omega\rvert - \lvert T\rvert + 1) \approx \lvert D\rvert \times \lvert\Omega\rvert$
$\Phi^{freq} = \sum_{i=1}^{m} q_i \geq \theta$	$O(mlog^2(min\_supp))$
$\Phi^{clos} = \bigwedge_{a \in \Omega} (p_a \vee \bigvee_{T_i \in D \mid a \notin T_i} q_i)$	$\lvert D\rvert - \lvert Supp(\{a\})\rvert$
$\Phi^{len} = \sum_{a \in \Omega} p_a \geq min\_length$	

Instance	θ	#Tran, #Items	Type of Data	#Clauses	#CFIM
Retail	10	88162, 16470	market basket data	1451119564	$> 1.10^5$
Kosarak	1000	990002, 41267	hungarian on-line news portal	40846393519	$\simeq 5.10^5$
Accidents	40000	340183, 468	traffic accidents	147704774	$\simeq 6.10^6$

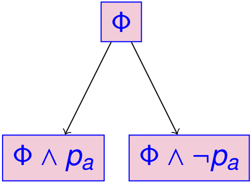
- **Scalability problem :** the number of clauses of the SAT encodings is very large.

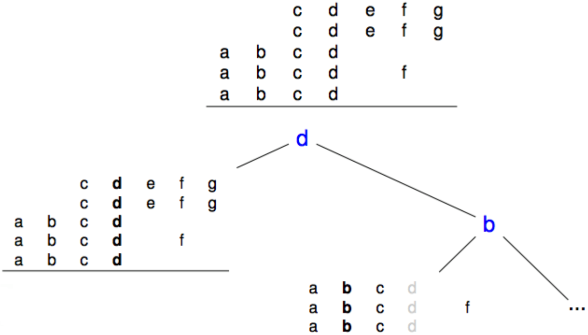
Decomposition-based SAT Approach for CFIM

$\Phi = \Phi^{cov} \wedge \Phi^{freq} \wedge \Phi^{clos}$, a : an item

$mod(\Phi \wedge p_a)$: itemsets with a

$mod(\Phi \wedge \neg p_a)$: itemsets without a





Decomposition & parallelism [PAKDD'14, CP'18]

Generate beforehand the set of guiding paths :

p_a
$\neg p_a \wedge p_b$
$\neg p_a \wedge \neg p_b \wedge p_c$
$\neg p_a \wedge \neg p_b \wedge \neg p_c \wedge p_d$
$\vdots$

$$\Phi \equiv \begin{array}{l} (\Phi \wedge p_a) \;\vee \\ (\Phi \wedge \neg p_a \wedge p_b) \;\vee \\ (\Phi \wedge \neg p_a \wedge \neg p_b \wedge p_c) \;\vee \\ (\Phi \wedge \neg p_a \wedge \neg p_b \wedge \neg p_c \wedge p_d) \;\vee \\ \vdots \end{array}$$

Items-based Guiding Paths Tree

Best policy : partition according to the items frequencies

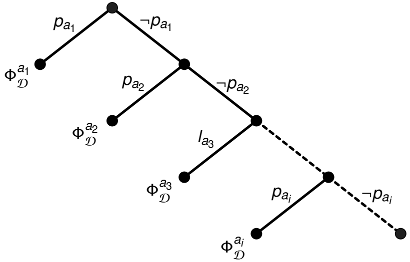
Decomposition & parallelism [PAKDD'14, CP'18]

Generate beforehand the set of guiding paths :

p_a
$\neg p_a \wedge p_b$
$\neg p_a \wedge \neg p_b \wedge p_c$
$\neg p_a \wedge \neg p_b \wedge \neg p_c \wedge p_d$
$\vdots$

$$\Phi \equiv \begin{array}{l} (\Phi \wedge p_a) \ \vee \\ (\Phi \wedge \neg p_a \wedge p_b) \ \vee \\ (\Phi \wedge \neg p_a \wedge \neg p_b \wedge p_c) \ \vee \\ (\Phi \wedge \neg p_a \wedge \neg p_b \wedge \neg p_c \wedge p_d) \ \vee \\ \vdots \end{array}$$

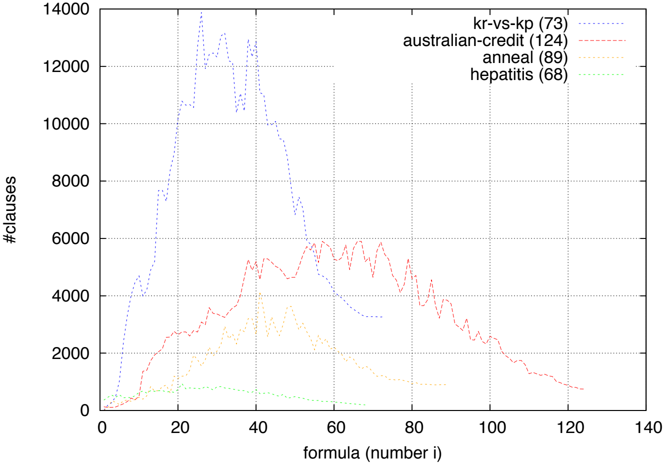
Items-based Guiding Paths Tree



Best policy : partition according to the items frequencies

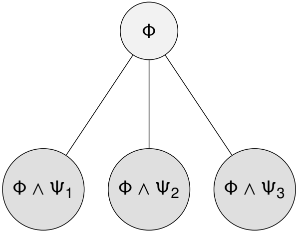
Decomposition-based SAT Approach for CFIM

Evolution of the number of clauses

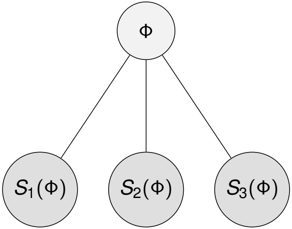


Main Parallel approaches

1. Divide and conquer approach
   - Divide the search space into sub-formulas, which are successively allocated to different SAT workers.



2. Portfolio-based approach
   - Let several differentiated engines compete and cooperate to be the first to solve a given instance.



paraSatMiner : A Parallel SAT Algorithm for CFIM

Algorithm 1: paraSatMiner

Input: $\mathcal{D}$, $\Omega = \{a_1, \ldots, a_m\}$, σ, θ, nb
Output: S : the set of Closed Fequent Itemsets

foreach $i \in \{1, \ldots, nb\}$ **do**
  $initEnumSatSolver(i)$;
  $\mathcal{M}_i = \emptyset$;
end
$\mathcal{S} = \emptyset$, $k = 0$;
# in parallel;
if $((i + k \times nb) \leq |\Omega|)$ **then**
  $\mathcal{M}_i \leftarrow \mathcal{M}_i \cup enumModels(enumSatSolver_i, \Phi_{\mathcal{D}}^{\sigma(a_{i+k\times nb})})$;
  $k{+}{+}$;
end
foreach $i \in \{1, \ldots, nb\}$ **do**
  $S = S \cup \mathcal{M}_i^{\mathcal{P}}$;
end
return S;

Experimental Evaluation

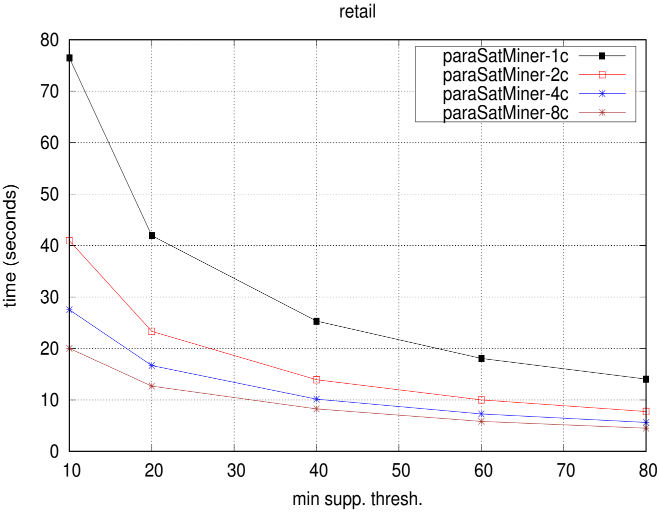
- OpenMP as an API that supports multi-platform shared memory multiprocessing
- Model enumeration solver based on MiniSAT
- Heuristic for Variable Selection : JW
- Intel Xeon quad-core machines with 32GB of RAM running at 2.66 Ghz
- Dense and sparse datasets (FIMI, CP4IM repositories)
- Timeout : 1000 seconds of CPU time

Sequential Evaluation

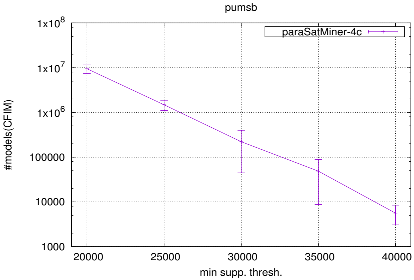
paraSatMiner vs (ClosedPattern, CoverSize, LCM)

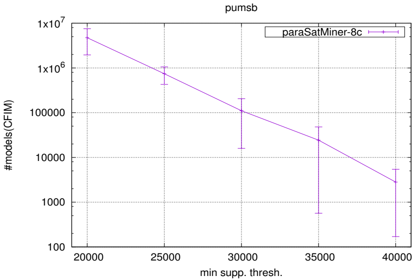
Instance	θ	Closed Pattern	Cover size	paraSat Miner-1c	LCM	Models
Retail	80	–	265.10	14.06	0.21	$> 8.10^3$
	60	–	295.47	18.07	0.24	$> 1.10^4$
	40	–	334.23	25.33	0.28	$> 2.10^4$
	20	–	439.94	41.93	0.35	$> 5.10^4$
	10	–	586.16	76.49	0.56	$> 1.10^5$
Chess	2000	1.51	1.22	0.25	0.04	$\simeq 7.10^4$
	1500	6.30	4.09	0.8	0.20	$> 5.10^5$
	1000	51.35	28.62	5.52	1.75	$> 4.10^6$
	500	577.29	311.47	49.50	18.25	$> 45.10^6$
	250	–	–	186.11	72.96	$\simeq 2.10^8$
	100	–	–	484.41	215.30	$> 5.10^8$

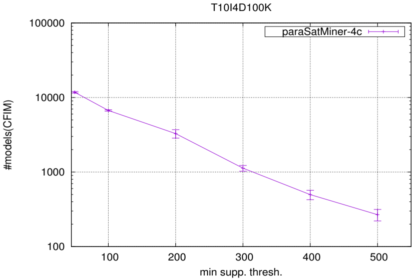
Parallel Evaluation

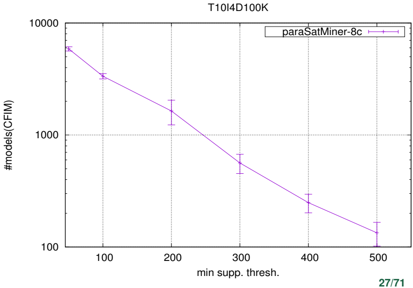


Load balancing analysis









SAT-based encodings for MFIM

- **Maximality constraint :**

$$\Phi^{max} = (\sum_{i=1...m, a \in T_i} q_i \geq \theta) \rightarrow p_a, \quad \text{for all } a \in \Omega$$

- **Maximal Itemsets Mining :**

$$\Phi^{max} \wedge \Phi^{cov} \wedge \Phi^{freq} \wedge \Phi^{clos}$$

Problem : The translation of the maximality constraint into CNF can lead to formula of huge size.

SAT-based encoding for MFIM

- To avoid encoding the maximality constraint :

1. Consider a DPLL-like procedure that selects the variables associated to items (p_a) first, and assign the value *true* first.

2. Add the following blocking clause to Φ, each time a model $\mathcal{B}$ is found :

$$C = (\bigvee_{a \in \Omega \setminus P(\mathcal{B})} p_a)$$

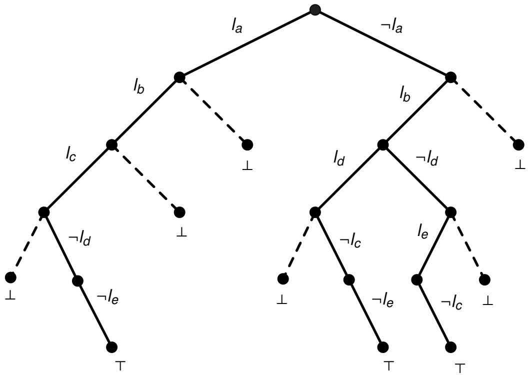
$\Longrightarrow$ The size of C can be considerably reduced as :

$$C = (\bigvee_{a \in T_i \setminus P(\mathcal{B})} p_a \vee \neg q_i)$$

SAT-based encoding for MFIM

TID	Transactions				
T_1	a	b	c	d	
T_2	a	b	c		e
T_3	a				e
T_4	a			d	e
T_5	a	b			
T_6		b		d	
T_7		b			e

Search Tree



$\mathcal{B} = \{p_a, p_b, p_c, \neg p_d, \neg p_e\}$

$C = (p_d \vee p_e) \Longrightarrow$ backtrack to the level 2

Experimental Evaluation

SATMax (+decomposition) vs (ECLAT, DMCP)

Instance	θ	ECLAT	DMCP	SATMax
Kosarak	3000	2.52	–	30.00
	2500	3.08	–	32.96
	2000	7.97	–	42.94
	1500	31.52	–	59.03
	1000	67.96	–	100.31
BMS-WebView-1	48	0.07	20.51	2.94
	36	0.22	195.68	5.56
	34	0.28	335.13	7.05
	32	0.36	553.39	7.43
	30	0.49	1049.28	7.14
Pumsb	40000	0.30	2.92	5.51
	35000	1.05	11.43	6.44
	30000	3.48	32.71	11.23
	25000	89.29	473	49.66
	20000	878.02	–	202.71

Association Rules Mining

Association Rule : A pattern $X \rightarrow Y$ s.t. X (antecedent) and Y (consequence) are two disjoint itemsets.

Support : $$Supp(X \rightarrow Y, D) = Supp(X \cup Y, D)$$

Confidence : $$Conf(X \rightarrow Y, D) = \frac{Supp(X \cup Y, D)}{Supp(X, D)}$$

$X \rightarrow Y$ is **closed** iff $X \cup Y$ is closed

Association Rules Mining problem : find the set $\{X \rightarrow Y \mid X, Y \subseteq \Omega,\ Supp(X \rightarrow Y) \geq \theta, Conf(X \rightarrow Y) \geq \lambda\}$

Association Rules Mining [IJCAI 2016]

$$\underbrace{\bigwedge_{a\in\Omega}(\neg x_a \vee \neg y_a)}_{X\cap Y=\emptyset} \quad \underbrace{\bigwedge_{i=1}^{m}(\neg p_i \leftrightarrow \bigvee_{a\in\Omega\setminus T_i} x_a)}_{Supp(X)} \quad \underbrace{\bigwedge_{i=1}^{m}(\neg q_i \leftrightarrow \neg p_i \vee (\bigvee_{a\in\Omega\setminus T_i} y_a))}_{Supp(X\cup Y)}$$

$$\underbrace{\sum_{i=1}^{m} q_i \geq \theta}_{Frequency} \quad \underbrace{\frac{\sum_{i=1}^{m} q_i}{\sum_{i=1}^{m} p_i} \geq \lambda}_{Confidence} \quad \underbrace{\bigwedge_{a\in\Omega}(\bigwedge_{a\notin T_i} q_i \rightarrow x_a \vee y_a)}_{Closeness}$$

Association Rules Mining : experiments

data (#items, #trans, density)	SFAR_Pure #S	SFAR_Pure avg. time(s)	ZART_Pure #S	ZART_Pure avg. time(s)	SFAR_Closed #S	SFAR_Closed avg. time(s)	ZART_Closed #S	ZART_Closed avg. time(s)
Audiology (148, 216, 45%)	20	855.00	20	855.01	20	855.00	20	855.01
Zoo-1 (36, 101, 44%)	400	19.12	400	6.37	400	0.52	400	11.28
Tic-tac-toe (27, 958, 33%)	400	0.09	400	0.24	400	0.09	400	0.23
Anneal (93, 812, 45%)	101	709.50	101	678.41	147	604.09	103	679.31
Australian-credit (125, 653, 41%)	245	370.17	264	321.62	268	323.29	226	403.72
German-credit (112, 1000, 34%)	306	246.88	322	192.52	329	198.02	304	238.79
Heart-cleveland (95, 296, 47%)	284	286.38	301	252.27	304	251.05	262	340.15
Hepatitis (68, 137, 50)	305	241.41	304	228.00	324	206.02	266	312.26
Hypothyroid (88, 3247, 49%)	85	732.12	121	665.41	107	686.95	64	761.59
Kr-vs-kp (73, 3196, 49%)	172	552.92	203	487.73	192	523.66	146	590.89
Lymph (68, 148, 40%)	336	181.64	338	170.37	387	63.22	291	281.35
Mushroom (119, 8124, 18%)	366	109.12	387	46.00	400	30.32	390	42.84
Primary-tumor (31, 336, 48%)	400	3.68	400	1.17	400	2.03	400	18.82
Soybean (50, 650, 32%)	400	2.90	400	1.50	400	0.17	400	7.94
Splice-1 (287, 3190, 21%)	380	53.44	400	3.52	380	54.04	400	3.25
Vote (48, 435, 33%)	380	66.74	400	1.46	400	32.40	398	30.22
Total	4560	279.76	**4741**	**247.29**	**4838**	**242.24**	4470	286.10

Non-redundant association rules [PAKDD'17]

Non redundant rule : if there is no $X' \rightarrow Y'$ different from $X \rightarrow Y$ s.t.

$$\left| \begin{array}{l} Supp(X \rightarrow Y) = Supp(X' \rightarrow Y'), \\ Conf(X \rightarrow Y) = Conf(X' \rightarrow Y'), \\ X' \subseteq X \text{ and } Y \subseteq Y' \end{array} \right.$$

Minimal Generator : $X' \subseteq X$ is a minimal generator of a **closed itemset** X iff

$$\left| \begin{array}{l} Supp(X') = Supp(X); \\ \text{There is no } X'' \subseteq X \text{ s.t. } X'' \subset X' \text{ and } Supp(X'') = Supp(X) \end{array} \right.$$

id	*transactions*					
1	*a*	*b*		*d*	*e*	*f*
2		**b**	**c**	**d**	*e*	*f*
3	*a*	**b**	**c**	**d**	*e*	*f*
4	*a*	**b**	**c**	**d**	*e*	*f*
5	*a*	*b*	*c*		*e*	
6			*c*	*d*		
7	*a*	*b*				

Non-redundant association rules

$X \to Y$ is a non-redundant rule iff X is a minimal generator ($|X| = 1$ or $\forall a \in X,\ Supp(X \setminus \{a\}) > Supp(X)$ and $X \cup Y$ is a closed itemset

$$x_a \to \sum_{b \in \Omega} x_b = 1 \quad \vee \quad \bigvee_{(i \in 1...m,\ a \notin T_i)} \Big(\bigwedge_{b \notin T_i \cup \{a\}} \neg x_b \Big)$$

$$x_a \to \underbrace{\sum_{b \in \Omega} x_b = 1}_{z_0} \quad \vee \quad \bigvee_{(i \in 1...m,\ a \notin T_i)} \underbrace{\Big(\sum_{b \notin T_i} x_b \leq 1 \Big)}_{z_i}$$

$$x_a \to z_0 \vee \bigvee_{(i \in 1...m,\ a \notin T_i)} z_i, \qquad z_0 \to \sum_{b \in \Omega} x_b = 1$$

$$\underbrace{\bigwedge_{((i \in 1...m,\ a \notin T_i)} z_i \to \sum_{b \notin T_i} x_b \leq 1)}_{cond.\ cardinality\ constraint [LPAR'18]}$$

Non-redundant association rules

$X \rightarrow Y$ is a non-redundant rule iff X is a minimal generator ($|X| = 1$ or $\forall a \in X,\ Supp(X \setminus \{a\}) > Supp(X)$ and $X \cup Y$ is a closed itemset

$$x_a \rightarrow \sum_{b \in \Omega} x_b = 1 \quad \vee \bigvee_{(i \in 1...m,\ a \notin T_i)} \Big(\bigwedge_{b \notin T_i \cup \{a\}} \neg x_b \Big)$$

$$x_a \rightarrow \underbrace{\sum_{b \in \Omega} x_b = 1}_{z_0} \quad \vee \bigvee_{(i \in 1...m,\ a \notin T_i)} \underbrace{\Big(\sum_{b \notin T_i} x_b \leq 1 \Big)}_{z_i}$$

$$x_a \rightarrow z_0 \vee \bigvee_{(i \in 1...m,\ a \notin T_i)} z_i, \qquad z_0 \rightarrow \sum_{b \in \Omega} x_b = 1$$

$$\bigwedge_{((i \in 1...m,\ a \notin T_i)} \underbrace{z_i \rightarrow \sum_{b \notin T_i} x_b \leq 1)}_{cond.\ cardinality\ constraint[LPAR'18]}$$

Non-redundant association rules

$X \rightarrow Y$ is a non-redundant rule iff X is a minimal generator ($|X| = 1$ or $\forall a \in X,\ Supp(X \setminus \{a\}) > Supp(X)$ and $X \cup Y$ is a closed itemset

$$x_a \rightarrow \sum_{b \in \Omega} x_b = 1 \quad \vee \quad \bigvee_{(i \in 1...m,\ a \notin T_i)} \Big(\bigwedge_{b \notin T_i \cup \{a\}} \neg x_b \Big)$$

$$x_a \rightarrow \underbrace{\sum_{b \in \Omega} x_b = 1}_{z_0} \quad \vee \quad \bigvee_{(i \in 1...m,\ a \notin T_i)} \underbrace{\Big(\sum_{b \notin T_i} x_b \leq 1 \Big)}_{z_i}$$

$$x_a \rightarrow z_0 \vee \bigvee_{(i \in 1...m,\ a \notin T_i)} z_i, \qquad z_0 \rightarrow \sum_{b \in \Omega} x_b = 1$$

$$\bigwedge_{((i \in 1...m,\ a \notin T_i)} \underbrace{z_i \rightarrow \sum_{b \notin T_i} x_b \leq 1)}_{cond.\ cardinality\ constraint[LPAR'18]}$$

Non-redundant association rules : experiments

	SAT4MNR-D		SAT4MNR		CORON	
data (#items, #trans, density)	#S	avg. time(s)	#S	avg. time(s)	#S	avg. time(s)
Audiology (148, 216, 45%)	21	854,82	21	854.87	20	855.01
Zoo-1 (36, 101, 44%)	400	0.23	400	0.27	400	1.35
Tic-tac-toe (27, 958, 33%)	400	0.34	400	0.14	400	0.24
Anneal (93, 812, 45%)	279	337.25	248	405.82	160	591.39
Australian-credit (125, 653, 41%)	298	265.74	278	309.32	251	352.01
German-credit (112, 1000, 34%)	354	149.03	328	212.58	321	206.34
Heart-cleveland (95, 296, 47%)	331	200.28	317	235.79	271	307.57
Hepatitis (68, 137, 50%)	360	140.69	343	170.89	286	284.09
Hypothyroid (88, 3247, 49%)	150	615.13	126	649.22	104	681.52
kr-vs-kp (73, 3196, 49%)	198	504.62	172	556.85	168	552.04
Lymph (68, 148, 40%)	400	6.78	400	19.21	357	131.07
Mushroom (119, 8124, 18%)	400	146.87	389	77.02	400	3.81
Primary-tumor (31, 336, 48%)	400	2.08	400	4.61	400	4.15
Soybean (50, 650, 32%)	400	0.36	400	0.20	400	0.61
Vote (48, 435, 33%)	400	5.43	400	30.46	364	87.56
Total	**4790**	**215.31**	**4622**	**235.15**	4302	270.58

FIM on Uncertain Transaction Databases

- Real-world data are often uncertain and imprecise
- An increasing application needs of handling a large amount of uncertain data
- Various applications : **sensor network monitoring, moving object search, object identification, etc.**
- Solutions for mining FIM over exact data cannot be directly applied to uncertain data
- Approximate methods have been proposed in the context of specialized approaches

FIM on Uncertain Transaction Databases

TID	Transactions						
T_1	$a(0.6)$	$b(0.3)$	$c(0.3)$	$d(0.5)$			
T_2	$a(0.6)$	$b(0.3)$	$c(0.8)$		$e(0.2)$		
T_3	$a(0.3)$	$b(0.8)$			$e(0.4)$		
T_4		$b(0.7)$		$d(0.3)$			
T_5						$f(0.2)$	$g(0.5)$

- **Uncertain transaction databases** $\mathcal{UD}$ **:** the probability of an item $a_j,\ (1 \leq j \leq m)$ in a transaction $T_i,\ (1 \leq i \leq n)$ is defined as :

$$p(a_j, T_i) = p_{ji}$$

FIM on Uncertain Transaction Databases

- The **existential probability** of an itemset I in T_i is defined :

$$p(I, T_i) = \prod_{a_j \in I, I \subseteq T_i} p_{ji}$$

- The **Expected Support Number** of an itemset $\mathcal{I}$ in $\mathcal{D}$ is defined :

$$ExpSN(I) = \sum_{T_i \in \mathcal{UD}} p(I, T_i)$$

The problem of **mining frequent itemset** over $\mathcal{UD}$ and a minimum support threshold θ is defined as :

$$FIM(\mathcal{UD}, \theta) = \{I \subseteq \Omega \mid ExpSN(I, \mathcal{UD}) \geq \theta\}$$

SAT Encoding of FIM over Uncertain Databases

- **Cover constraint :**

$$\Phi^{cov} = \bigwedge_{i=1}^{n} (\neg q_i \leftrightarrow \bigvee_{a \in \Omega \setminus T_i} p_a)$$

- **Frequency constraint :**

$$\Phi^{freq} = \sum_{i=1}^{n} \prod_{a \in T_i} (p(a, T_i) \times p_a \wedge q_i + \neg p_a \wedge q_i) \geq \theta$$

- FIM($\mathcal{UD}$) : $\Phi^{fim} = \Phi^{cov} \wedge \Phi^{freq}$

The translation of the frequency constraint into a linear one is intractable.

Relaxation-based Computation of FIM

- The **maximum** existential probability :

$$p_{\max}(I, T_i) = p_{\max}(k, T_i) = \max_{|J|=k} \prod_{a \in J,\ \mathcal{J} \subseteq T_i} p(a, T_i)$$

- The **relaxed** expected support number of $\mathcal{I}$:

$$R\_ExpSN(I, \mathcal{UD}) = \sum_{T_i \in \mathcal{UD}} p_{max}(I, T_i)$$

$$R\_ExpSN(I, \mathcal{UD}) \geq ExpSN(I, \mathcal{UD})$$

Relaxation based computation of FIM

$$\Phi^k = (\sum_{a \in \Omega} p_a = k) \wedge (\sum_{T_i \in \mathcal{UD}} p_{max}(k, T_i) \times q_i \geq \theta)$$

Algorithm 2: Iterative SAT-based Itemsets Enumeration

Input: An Uncertain Transaction Database $\mathcal{UD}$
Output: The set of all frequent itemsets $\mathcal{S}$

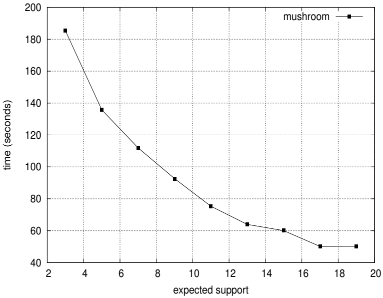
$\mathcal{G} \leftarrow \mathit{SATEncodingTable}(\mathcal{D})$;
$\mathcal{S} \leftarrow \emptyset$; $\mathcal{S}' \leftarrow \emptyset$; `/* Boolean models */`
$k \leftarrow 0$; `/* size of itemsets */`
repeat
  $k \leftarrow k + 1$;
  $\mathcal{S}' \leftarrow \mathit{enumModels}(\Phi^k \wedge \mathcal{G})$;
  $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}'$;
until $(\mathcal{S}' = \emptyset)$;
return $\mathcal{S}$;

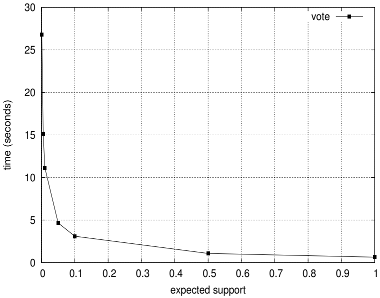
Performance Evaluation

The average of false positives

Dataset	θ	% False Positives
zoo_1	0.1	30.29
tic-tac-toe	0.1	4.84
vote	0.1	31.50
soybean	0.1	30.52
primary_tumor	0.1	25.08

Results by varying the support value





Gradual Itemsets Mining

Graduality

- Represents variation between elements
- **"the more X is A, the more Y is B"**
- Initially used in the fuzzy domain
  - expert systems

Example

- **The more experience, the higher salary**
- **The older a subject, the less his memory**

Various applications :

- **Medecine**: correlations between memory and feeling points
- **Biology**: correlations between genomic expressions

Gradual patterns

Object	(P)	(S)	(R)
t_1	0	0	5
t_2	31	7	3
t_3	62	8	9
t_4	18	1	0
t_5	13	1	4
t_6	17	2	1
t_7	36	3	6

Gradual item is denoted i^* with $* \in \{+, -\}$

- $* = +$ means the value of i is increasing and $* = -$ means the value of i is decreasing

What is the variation ?

- $+$ corresponds to $\geq$ and $-$ corresponds to $\leq$
- As we are **comparing objects**, order is expressed as :
  - $t_1[i] \leq t_2[i]$, then we write i^+
  - $t_1[i] \geq t_2[i]$, then we write i^-

Gradual patterns

Object	(P)	(S)	(R)
t_1	0	0	5
t_2	31	7	3
t_3	62	8	9
t_4	18	1	0
t_5	13	1	4
t_6	17	2	1
t_7	36	3	6

Gradual item is denoted i^* with $* \in \{+, -\}$

- $* = +$ means the value of i is increasing and $* = -$ means the value of i is decreasing

Example: P^+

Object	(P)
t_1	0
t_2	31
t_3	62
t_4	18
t_5	13
t_6	17
t_7	36

Object	(P)
t_1	0
t_5	13
t_6	17
t_4	18
t_2	31
t_7	36
t_3	62

Gradual item, Gradual pattern

Gradual pattern (itemset)

$g = (i_1^{*1}, ..., i_k^{*k})$ is a non empty set of gradual items.

Example: **(P^+, R^-)**

Object	(P)	(R)
t_1	0	5
t_5	13	4
t_6	17	1
t_4	18	0

Complementary gradual itemset

- $g = (i_1^{*1}, ..., i_k^{*k})$ and c such that "$c(\geq) = \leq$" and "$c(\leq) = \geq$"
- $c(g)$ denotes the complementary gradual itemset of g
- Example : $c(P^+ S^+ R^-) = P^- S^- R^+$

Gradual Pattern Extension

Gradual Pattern Extension

- Let $g = (i_1^{*_1}, ..., i_k^{*_k})$ be a gradual itemset
- Let $s = \langle t_1 \rightarrow t_2 \rightarrow \ldots \rightarrow t_n \rangle$ be a sequence of tuples

s is an extension of g if, $\forall\ 1 \leq p \leq k$ and $\forall\ 1 \leq j < n$, the following constraint is satisfied :

$$t_j[i_p] \ *_p \ t_{j+1}[i_p]$$

Cover

- Let $g = (i_1^{*_1}, ..., i_k^{*_k})$ be a gradual itemset of a database Δ.

Then, $Cover(g, \Delta)$ is the set of the longest extensions of g in Δ with respect to set inclusion.

Gradual Itemset Support

- Let Δ be a numerical database and g be a gradual itemset of Δ.

$$Supp(g, \Delta) = \frac{max\{|s|,\ s \in Cover(g, \Delta)\}}{|\Delta|}$$

Object	(P)	(S)	(R)
t_1	0	0	5
t_2	31	7	3
t_3	62	8	9
t_4	18	1	0
t_5	13	1	4
t_6	17	2	1
t_7	36	3	6

- $g = (P^+, R^-)$
- $Cover(g, \Delta) = \{\langle t_1, t_5, t_2\rangle, \langle t_1, t_5, t_6, t_4\rangle\}$
- $Supp(s) = \frac{4}{7} = 0.57$ (57%)
- $Supp(i^*) = 100\%$
- g is frequent if its support is higher than a given support threshold

Frequent Gradual Itemsets Mining Problem

Definition

- Let Δ be a numerical database
- Let θ be a minimum support threshold

The problem of mining gradual itemsets is to find the set of all frequent gradual itemsets of Δ with respect to θ.

Motivation

Limits of the state-of-the-art approaches

- Generate a unique extension for each frequent gradual itemset
  - **all the extensions might be required to explain the gradualness of patterns or to derive additional knowledge**
- Do not take into account equality between attribute values
  - let $g = (a^{\geq}, b^{\geq})$ be a gradual itemset and $\langle t_1 \rightarrow \ldots \rightarrow t_m \rangle$ its associated extension
  - g is valid even if : $t_1[a] < \ldots < t_m[a]$, and $t_1[b] = \ldots = t_{i+1}[b]$

Our aim

- Enumerate all extensions associated to each gradual pattern
- Take into account the equality case
- **Exploit existing sequence mining algorithms**

Motivation

Valid Gradual Pattern Extension

- Let $g = (i_1^{*1}, ..., i_k^{*k})$ be a gradual itemset
- An extension $s = \langle t_1 \to t_2 \to \ldots \to t_n \rangle$ of g is valid if $\forall\ 1 \leq j < n$, and $\forall\ 1 \leq p < q \leq k$,

$$t_j[i_p] = t_{j+1}[i_p] \text{ iff } t_j[i_q] = t_{j+1}[i_q] \quad (1)$$

Numerical Database

Object	age	salary	cars
t_1	22	1200	2
t_2	28	1850	3
t_3	24	1200	4
t_4	35	2200	4
t_5	38	2000	1
t_6	44	3400	1
t_7	52	3400	3
t_8	41	5000	2

$g = (age^{\geq}, salary^{\geq})$

- $\langle t_3 \to t_2 \to t_4 \to t_6 \rangle$ is a valid extension associated to g
- $\langle t_1 \to t_3 \to t_2 \to t_4 \to t_6 \to t_7 \rangle$ is not a valid extension associated to g

Gradual Patterns Mining as Sequence Mining [FUZZ-IEEE'2019]

Let Δ be a numerical database over a set of numerical attributes $\mathcal{A} = \{i_1, \ldots, i_m\}$ and objects $\mathcal{T} = \{t_1, \ldots, t_n\}$. Given a gradual item i^* with $i \in \mathcal{A}$, we define G_i^* as the sequence of objects $\langle t_1 \rightarrow \ldots \rightarrow t_n \rangle$ satisfying i^*

Object	**age**	**salary**	**cars**
t_1	22	1200	2
t_2	28	1850	3
t_3	24	1200	4
t_4	35	2200	4
t_5	38	2000	1
t_6	44	3400	1
t_7	52	3400	3
t_8	41	5000	2

- $G^{\geq}_{salary} = \langle t_1 t_3 \rightarrow t_2 \rightarrow t_5 \rightarrow t_4 \rightarrow t_6 t_7 \rightarrow t_8 \rangle$
- A given i^* corresponds to a unique sequence G_i^* of itemsets

Gradual Patterns Mining as Sequence Mining

Let Δ be a numerical database. We define $\delta(\Delta)$ as

$$\delta(\Delta) = \{(i_1^{\geq}, G_{i_1}^{\geq}), (i_1^{\leq}, G_{i_1}^{\leq}), \ldots, (i_m^{\geq}, G_{i_m}^{\geq}), (i_m^{\leq}, G_{i_m}^{\leq})\}$$

Object	**age**	**salary**	**cars**
t_1	22	1200	2
t_2	28	1850	3
t_3	24	1200	4
t_4	35	2200	4
t_5	38	2000	1
t_6	44	3400	1
t_7	52	3400	3
t_8	41	5000	2

Gradual Items	**Sequences**
$age^{\geq}$	$\langle t_1 \rightarrow t_3 \rightarrow t_2 \rightarrow t_4 \rightarrow t_5 \rightarrow t_8 \rightarrow t_6 \rightarrow t_7 \rangle$
$age^{\leq}$	$\langle t_7 \rightarrow t_6 \rightarrow t_8 \rightarrow t_5 \rightarrow t_4 \rightarrow t_2 \rightarrow t_3 \rightarrow t_1 \rangle$
$salary^{\geq}$	$\langle t_1 t_3 \rightarrow t_2 \rightarrow t_5 \rightarrow t_4 \rightarrow t_6 t_7 \rightarrow t_8 \rangle$
$salary^{\leq}$	$\langle t_8 \rightarrow t_6 t_7 \rightarrow t_4 \rightarrow t_5 \rightarrow t_2 \rightarrow t_1 t_3 \rangle$
$cars^{\geq}$	$\langle t_5 t_6 \rightarrow t_8 t_1 \rightarrow t_2 t_7 \rightarrow t_4 t_3 \rangle$
$cars^{\leq}$	$\langle t_4 t_3 \rightarrow t_2 t_7 \rightarrow t_8 t_1 \rightarrow t_5 t_6 \rangle$

A Sequence Mining Approach for Mining Gradual Patterns

Gradual Items	Sequences
$age^{\geq}$	$\langle t_1 \rightarrow t_3 \rightarrow t_2 \rightarrow t_4 \rightarrow t_5 \rightarrow t_8 \rightarrow t_6 \rightarrow t_7 \rangle$
$age^{\leq}$	$\langle t_7 \rightarrow t_6 \rightarrow t_8 \rightarrow t_5 \rightarrow t_4 \rightarrow t_2 \rightarrow t_3 \rightarrow t_1 \rangle$
$salary^{\geq}$	$\langle t_1 t_3 \rightarrow t_2 \rightarrow t_5 \rightarrow t_4 \rightarrow t_6 t_7 \rightarrow t_8 \rangle$
$salary^{\leq}$	$\langle t_8 \rightarrow t_6 t_7 \rightarrow t_4 \rightarrow t_5 \rightarrow t_2 \rightarrow t_1 t_3 \rangle$
$cars^{\geq}$	$\langle t_5 t_6 \rightarrow t_8 t_1 \rightarrow t_2 t_7 \rightarrow t_4 t_3 \rangle$
$cars^{\leq}$	$\langle t_4 t_3 \rightarrow t_2 t_7 \rightarrow t_8 t_1 \rightarrow t_5 t_6 \rangle$

Lemma

if $\langle t_1 \rightarrow t_2 \ldots \rightarrow t_n \rangle$ is frequent sequence in $\delta(\Delta)$, then $\langle t_n \rightarrow t_{n-1} \rightarrow \ldots \rightarrow t_1 \rangle$ is also a frequent sequence.

Experiments

- A real world database about paleoecological data containing 111 objects and 40 attributes

θ	**#Grad_cl**	**#Grad.** (**#Ext.**)	**time (s)**
0.20	21 941 457	**598 655 (2 067 533)**	23875.90
0.25	10 186 219	**252 441 (876 39)**	12834.10
0.30	4 747 460	**121 864 (531 978)**	7267.12
0.40	1 098 143	**76 532 (267 861)**	1761.27
0.45	407 625	**49 234 (94 591)**	629.78
0.50	130 172	**21 563 (61 793)**	216.86
0.60	12 218	**5 099 (3 768)**	22.26
0.70	778	1 078 (879)	1.95
0.80	130	**99 (80)**	0.47
0.90	51	53 (43)	0.23

- Reduce considerably the number of gradual itemsets
- Computation time increases when the support threshold decreases

A SAT-Based Model for Mining Gradual Patterns

- $\mathcal{A} = \{a_1, \ldots, a_m\}$: a set of attributes
- $\mathcal{T} = \{t_1, \ldots, t_n\}$: a set of objects
- $\mathcal{A}^* = \{a_1^+, a_1^-, \ldots, a_m^+, a_m^-\}$: the set of attribute variations
- k : the minimum support threshold

- Associate to each attribute $a \in \mathcal{A}$ two boolean variables respectively x_{a^+} and x_{a^-}
  - Such boolean variables encode the candidate itemset g, i.e, $x_{a^*} = true$ **iff** $a^* \in g$

- Let $\langle t_1 \rightarrow \ldots \rightarrow t_k \rangle$ be the longest sequence of objects required for a frequent gradual itemset
  - Associate boolean variable y_{ij} to express that object t_i is putted in the position j

A SAT-Based Model for Mining Gradual Patterns

- A constraint to capture consistency of the candidate gradual itemset (does not contain both a^+ and a^-):

$$\bigwedge_{a \in a_1 \ldots a_m} (\neg x_{a^+} \vee \neg x_{a^-})$$

- A constraint to place uniquely one object t_i in the *jth* position of the gradual itemset extension:

$$\bigwedge_{1 \leq j \leq k} (\sum_{i=1}^{n} y_{ij} = 1)$$

- A constraint to prevent an object to be placed in more than one position of the gradual itemset extension:

$$\bigwedge_{1 \leq i \leq n} (\sum_{j=1}^{k} y_{ij} \leq 1)$$

SAT-based Encoding for Mining Frequent Gradual Patterns

- A constraint that expresses for a given gradual item $a^\diamond$, the set of objects that can be set in position $j+1$:

$$\bigwedge_{a^\diamond \in \mathcal{A}^*} \bigwedge_{1 \leq i \leq n} \bigwedge_{1 \leq j \leq k} (x_{a^\diamond} \wedge y_{ij} \rightarrow \bigvee_{t_k[a] \,\diamond\, t_i[a]} y_{k(j+1)})$$

- Can be expressed differently:

$$\bigwedge_{a^\diamond \in \mathcal{A}^*} \bigwedge_{1 \leq i \leq n} \bigwedge_{1 \leq j \leq k} (x_{a^\diamond} \wedge y_{ij} \rightarrow \bigwedge_{t_k[a] \,\overline{\diamond}\, t_i[a]} \neg y_{k(j+1)})$$

- Eliminate symmetrical gradual itemsets

SAT Based Gradual Patterns Enumeration

Experiments

- Implemented in *Minisat2.2* without **learning clause**
- Dataset: 100 objects and 10 attributes

#minSupp (%)	#Vars	#Clauses	#Gradual model	Time (seconds)
5	1 419	337 516	24 468	97.19
10	2 914	759 151	4 362	391.43
15	4 409	1 180 786	2 404	3518.47
20	5 904	1 602 421	459	11637.5
25	7 399	2 024 056	214	29578.36
30	8 894	2 445 691	144	38210.58
35	10 389	2 867 326	82	55480.58
40	11 884	3 288 961	58	60480.58
45	13 379	3 710 596	46	-
50	14 874	4 132 231	20	-

TABLE – Characteristics of instances & Enumeration Time

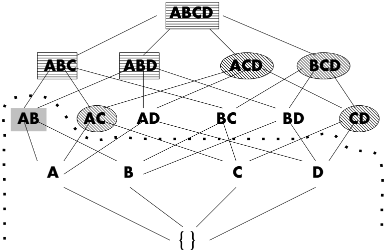
Symmetries [ECAI'12, ICTAI'13]

Symmetry : permutation σ over Ω such that $\sigma(D) = D$
It can be represented as a set of cycles :

$$\sigma = (a_1, b_1)(a_2, b_2)\ldots(a_n, b_n)$$

Symmetry breaking

1. Preprocessing : remove b_i from each transaction not involving $\{a_1, \ldots, a_i\}$
2. During search : use symmetry breaking during candidates generation for Apriori-based algorithms



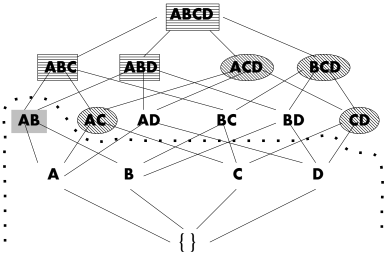
Symmetries [ECAI'12, ICTAI'13]

Symmetry : permutation σ over Ω such that $\sigma(D) = D$
It can be represented as a set of cycles :

$$\sigma = (a_1, b_1)(a_2, b_2)\dots(a_n, b_n)$$

Symmetry breaking

1. Preprocessing : remove b_i from each transaction not involving $\{a_1, \dots, a_i\}$
2. During search : use symmetry breaking during candidates generation for Apriori-based algorithms



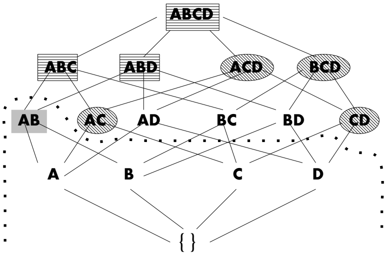
Symmetries [ECAI'12, ICTAI'13]

Symmetry : permutation σ over Ω such that $\sigma(D) = D$
It can be represented as a set of cycles :

$$\sigma = (a_1, b_1)(a_2, b_2) \ldots (a_n, b_n)$$

Symmetry breaking

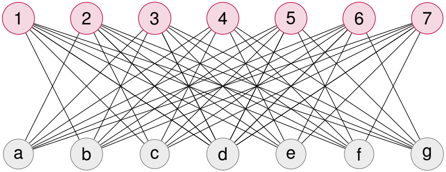
1. Preprocessing : remove b_i from each transaction not involving $\{a_1, \ldots, a_i\}$
2. During search : use symmetry breaking during candidates generation for Apriori-based algorithms



Itemsets Mining & Symmetries [ECAI'12]

Symmetry Breaking as a preprocessing step

id	*transactions*						
1		*b*	*c*	*d*	*e*	*f*	*g*
2	*a*		*c*	*d*	*e*	*f*	*g*
3	*a*	*b*		*d*	*e*	*f*	*g*
4	*a*	*b*	*c*		*e*	*f*	*g*
5	*a*	*b*	*c*	*d*		*f*	*g*
6	*a*	*b*	*c*	*d*	*e*		*g*
7	*a*	*b*	*c*	*d*	*e*	*f*	



id	*transactions*						
1		*b*	*c*	*d*	*e*	*f*	*g*
2	*a*		*c*	*d*	*e*	*F*	*g*
3	*a*	*b*		*d*	*e*	*f*	*g*
4	*a*	*b*	*c*		*e*	*f*	*g*
5	*a*	*b*	*c*	*d*		*f*	*g*
6	*a*	*b*	*c*	*d*	*e*		*g*
7	*a*	*b*	*c*	*d*	*e*	*f*	

$$\sigma_1 = (a, b) \quad \sigma_2 = (b, c)$$
$$\sigma_3 = (c, d) \quad \sigma_4 = (d, e)$$
$$\sigma_5 = (e, f) \quad \sigma_6 = (f, g)$$

CNF Formulas compression [CIKM'13]

Big Formulas : continuous challenge of SAT solving

$(\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_6)$

$(x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_1 \vee x_4) \wedge (x_1 \vee x_5) \wedge$

$(x_2 \vee x_3) \wedge (x_2 \vee x_4) \wedge (x_2 \vee x_5) \wedge$

$(x_3 \vee x_4) \wedge (x_3 \vee x_5) \wedge$

$(x_4 \vee x_5)$

Itemsets Mining + Tseitin principle

$(y_1 \vee x_4) \wedge (y_1 \vee x_5) \wedge (y_1 \vee x_6)$

$(\neg y_1 \vee \neg x_1 \vee \neg x_2 \vee \neg x_3)$

$(x_1 \vee x_6 \wedge x_5 \wedge x_4 \wedge x_3 \wedge x_2)$

$(x_2 \vee x_6 \wedge x_5 \wedge x_4 \wedge x_3)$

$(x_3 \vee x_6 \wedge x_5 \wedge x_4)$

$(x_4 \vee x_6 \wedge x_5)$

$(x_5 \vee x_6)$

$(y_1 \vee x_4) \wedge (y_1 \vee x_5) \wedge (y_1 \vee x_6)$

$(\neg y_1 \vee \neg x_1 \vee \neg x_2 \vee \neg x_3)$

$(x_1 \vee y_2 \wedge x_4 \wedge x_3 \wedge x_2)$

$(x_2 \vee y_2 \wedge x_4 \wedge x_3)$

$(x_3 \vee y_2)$

$(x_4 \vee y_2)$

$(x_5 \vee x_6)$

$(\neg y_2 \vee x_6 \wedge x_5)$

CNF Formulas compression [CIKM'13]

Big Formulas : continuous challenge of SAT solving

$$(\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_6)$$
$$(x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_1 \vee x_4) \wedge (x_1 \vee x_5) \wedge$$
$$(x_2 \vee x_3) \wedge (x_2 \vee x_4) \wedge (x_2 \vee x_5) \wedge$$
$$(x_3 \vee x_4) \wedge (x_3 \vee x_5) \wedge$$
$$(x_4 \vee x_5)$$

Itemsets Mining + Tseitin principle

$$(y_1 \vee x_4) \wedge (y_1 \vee x_5) \wedge (y_1 \vee x_6)$$
$$(\neg y_1 \vee \neg x_1 \vee \neg x_2 \vee \neg x_3)$$
$$(x_1 \vee x_6 \wedge x_5 \wedge x_4 \wedge x_3 \wedge x_2)$$
$$(x_2 \vee x_6 \wedge x_5 \wedge x_4 \wedge x_3)$$
$$(x_3 \vee x_6 \wedge x_5 \wedge x_4)$$
$$(x_4 \vee x_6 \wedge x_5)$$
$$(x_5 \vee x_6)$$

$$(y_1 \vee x_4) \wedge (y_1 \vee x_5) \wedge (y_1 \vee x_6)$$
$$(\neg y_1 \vee \neg x_1 \vee \neg x_2 \vee \neg x_3)$$
$$(x_1 \vee y_2 \wedge x_4 \wedge x_3 \wedge x_2)$$
$$(x_2 \vee y_2 \wedge x_4 \wedge x_3)$$
$$(x_3 \vee y_2)$$
$$(x_4 \vee y_2)$$
$$(x_5 \vee x_6)$$
$$(\neg y_2 \vee x_6 \wedge x_5)$$

CNF Formulas compression [CIKM'13]

Big Formulas : continuous challenge of SAT solving

$(\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_5) \wedge (\neg x_1 \vee \neg x_2 \vee \neg x_3 \vee x_6)$
$(x_1 \vee x_2) \wedge (x_1 \vee x_3) \wedge (x_1 \vee x_4) \wedge (x_1 \vee x_5) \wedge$
$(x_2 \vee x_3) \wedge (x_2 \vee x_4) \wedge (x_2 \vee x_5) \wedge$
$(x_3 \vee x_4) \wedge (x_3 \vee x_5) \wedge$
$(x_4 \vee x_5)$

Itemsets Mining + Tseitin principle

$(y_1 \vee x_4) \wedge (y_1 \vee x_5) \wedge (y_1 \vee x_6)$
$(\neg y_1 \vee \neg x_1 \vee \neg x_2 \vee \neg x_3)$
$(x_1 \vee x_6 \wedge x_5 \wedge x_4 \wedge x_3 \wedge x_2)$
$(x_2 \vee x_6 \wedge x_5 \wedge x_4 \wedge x_3)$
$(x_3 \vee x_6 \wedge x_5 \wedge x_4)$
$(x_4 \vee x_6 \wedge x_5)$
$(x_5 \vee x_6)$

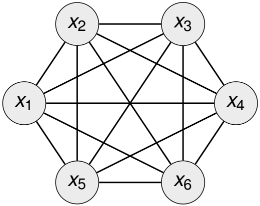
$(y_1 \vee x_4) \wedge (y_1 \vee x_5) \wedge (y_1 \vee x_6)$
$(\neg y_1 \vee \neg x_1 \vee \neg x_2 \vee \neg x_3)$
$(x_1 \vee y_2 \wedge x_4 \wedge x_3 \wedge x_2)$
$(x_2 \vee y_2 \wedge x_4 \wedge x_3)$
$(x_3 \vee y_2)$
$(x_4 \vee y_2)$
$(x_5 \vee x_6)$
$(\neg y_2 \vee x_6 \wedge x_5)$

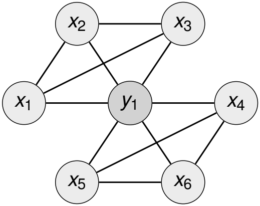
CNF Formulas compression [CIKM'13]

$$\Phi_{\leq 1}(x_1, \ldots, x_n) = \sum_{i=1}^{n} \neg x_i \leq 1 = \bigwedge_{1 \leq i < j \leq n} (x_i \vee x_j)$$

$$\left[\begin{array}{l} x_1 \vee x_6 \wedge x_5 \wedge x_4 \wedge x_3 \wedge x_2 \\ x_2 \vee x_6 \wedge x_5 \wedge x_4 \wedge x_3 \\ x_3 \vee x_6 \wedge x_5 \wedge x_4 \\ x_4 \vee x_6 \wedge x_5 \\ x_5 \vee x_6 \end{array}\right]$$

$$\left[\begin{array}{l} x_1 \vee y_1 \wedge x_3 \wedge x_2 \\ x_2 \vee y_1 \wedge x_3 \\ x_3 \vee y_1 \\ \hline \neg y_1 \vee x_6 \wedge x_5 \wedge x_4 \\ x_4 \vee x_6 \wedge x_5 \\ x_5 \vee x_6 \end{array}\right]$$





$$\Phi_{\leq 1}(x_1, \ldots, x_n) = \Phi_{\leq 1}(x_1, \ldots, x_{\frac{n}{2}}, b) \wedge \Phi_{\leq 1}(\neg b, x_{\frac{n}{2}+1}, \ldots, x_n)$$

Graphs summarization

Interests :

- Store large graphs in memory
- Visualize graphs to more understand their structures
- Make efficiently computations on graphs

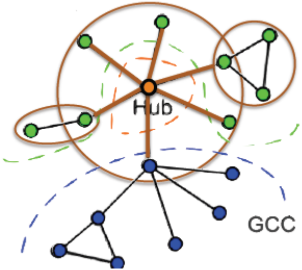
Limitations :

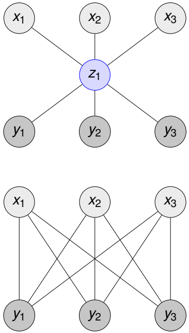
- Important structural properties
- High complexity
- Scalability

Graphs summarization

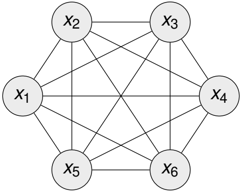
existing approaches :

- Node-based [Zhou et al .10]
- Edge-based [Francisco et al .07]
- Structure-based [Koutra et al .14]

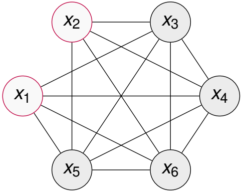




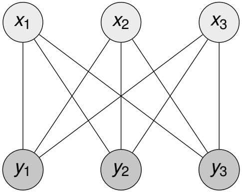
Graphs summarization [BigData'16]



Clique



Quasi-Clique



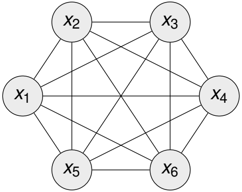
Bipartite complete

$$\sum_{i=1}^{n} x_i = 2 \qquad x_1 + x_2 + \sum_{i=3}^{n} 2x_i \geq 3 \qquad \sum_{i=1}^{n} 2x_i + \sum_{i=1}^{m} 3y_i = 5$$

2-models of PB constraints are edges of the corresponding graphs

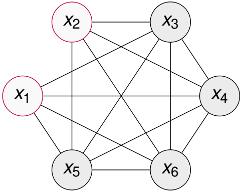
Look for $G'(V' \cup V'', E') \subseteq G(V, E)$ that can be modeled as a PB constraint

Graphs summarization [BigData'16]



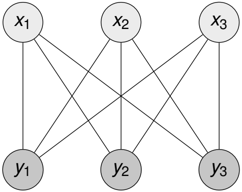
Clique

$$\sum_{i=1}^{n} x_i = 2$$



Quasi-Clique

$$x_1 + x_2 + \sum_{i=3}^{n} 2x_i \geq 3$$



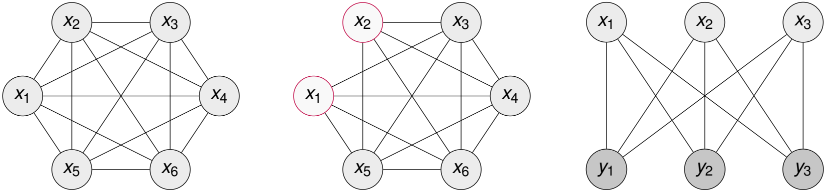
Bipartite complete

$$\sum_{i=1}^{n} 2x_i + \sum_{i=1}^{m} 3y_i = 5$$

2-models of PB constraints are edges of the corresponding graphs

Look for $G'(V' \cup V'', E') \subseteq G(V, E)$ that can be modeled as a PB constraint

Graphs summarization [BigData'16]



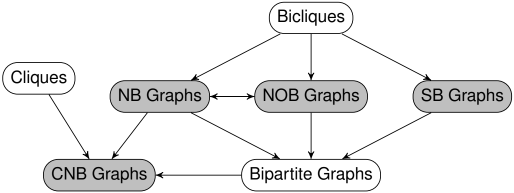
Clique	Quasi-Clique	Bipartite complete
$\sum_{i=1}^{n} x_i = 2$	$x_1 + x_2 + \sum_{i=3}^{n} 2x_i \geq 3$	$\sum_{i=1}^{n} 2x_i + \sum_{i=1}^{m} 3y_i = 5$

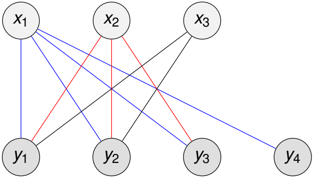
2-models of PB constraints are edges of the corresponding graphs

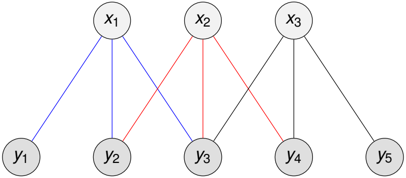
Look for $G'(V' \cup V'', E') \subseteq G(V, E)$ that can be modeled as a PB constraint

Graphs summarization [BigData'16]

- Nested Bipartite Graphs (NB), Clique Nested Bipartite Graphs (CNB), Sequence Bipartite graphs (SB)







$$0 \leq \sum_{i=1}^{n} (m + m_i)\, x_i - \sum_{j=1}^{m} (m + j)\, y_j \leq m$$

$$1 \leq \sum_{j=1}^{m} (k + j)\, y_j - \sum_{i=1}^{n} (k + k_i)\, x_i \leq k$$

Experimental Evaluation

Compression performance (VOG vs SuLI) :

Graph	#nodes/#edges	size	#NB	time (s)	Compression Rate	
					VOG (%)	SuLI (%)
Chocolate	4 039/87 885	940.3KB	57	9 654	39.14	64.14
Facebook	473 315/3 505 519	47MB	12 800	501.94	68.08	62.97
Ca-AstroPh	18 772/198 110	207.7KB	3 119	340	25	**27.78**
Twitter	18 772/198 050	4MB	3119	309.6	65	**75.14**
Enron	36 691/186 936	4MB	718	8 754	32.5	**47.5**
epinions	75 877/405 739	380.4KB	924	1 387	60.63	47
Cit-hep-th	27 400/352 021	658.6KB	9 388	1 765	67.07	**82.02**
cnr-2000	325 557/3 216 152	41.5MB	487	417	39.03	**40.24**
DBLP	317 080/1 049 866	13.4MB	8 281	5 785	19.40	14.92
LiveJournal	3 997 962/34 681 189	50.4MB	4 365	3 643	80	67.46
Youtube	1 134 890/2 987 625	38.2MB	8 000	2 111.4	13.08	**30.36**
Flickr	105 938/2 316 948	48.7MB	8 084	4 837	59.54	39.01
Yahoo	105 938/2 316 948	24.9MB	4 800	6 511	48.99	**54.61**

Conclusion & Perspectives

Conclusion

1. Efficient encodings (declarative) for many data mining tasks

2. Decomposition and Parallel approches to tackle large data

3. Cross-fertilization between AI and Data mining (Symmetries, Compression)