Symétries et Extraction de Motifs Ensemblistes

Saïd Jabbour, Mehdi Khiari, Lakhdar Saïs, Yakoub Salhi and Karim Tabia

CRIL - CNRS UMR 8188
Université d'Artois, France

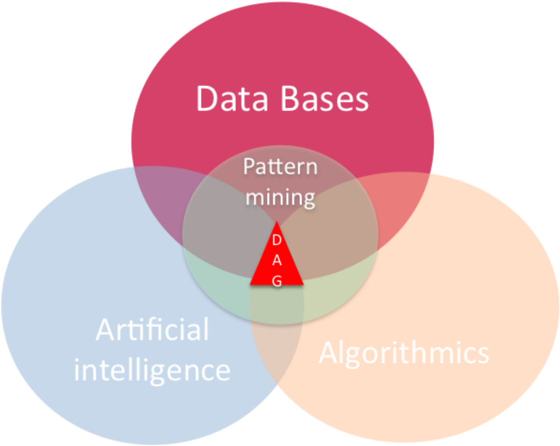
Work done in the framework of Project DAG ANR Défis 2009

January 30, 2014

Outline

ANR DEFI 2009: DAG

DAG: Declarative Approaches for Enumerating Interesting Patterns



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

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

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.

Symmetries in CP and SAT

- Symmetry resolution proof system [Krishnamurthy'85]
- Dynamic symmetry detection and elimination in propositional calculus [Benhamou et al. 92]
- Interchangeability [Freuder'91]. Variable and value symmetries [Puget'93]
- Symmetry breaking predicates [Crawford'92, Puget'93]
- Many other contributions (e.g. [Walsh'2012, Karem A. Sakallah'2011] ...)

Outline

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 ($0 < \lambda \leq |\mathcal{D}|$), the frequent itemset mining problem consists in computing the set of itemsets

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

Definition (Frequent Itemset Mining Problem)

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

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

Proposition (Anti-Monotonicity)

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

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

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 $\sigma(\mathcal{D}) = \{\sigma(t_i, I) = (\sigma(t_i), \sigma(I)), (t_i, I) \in \mathcal{D}\}$

Definition (Permutation)

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

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 $\sigma(a_k) = a_1$.

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

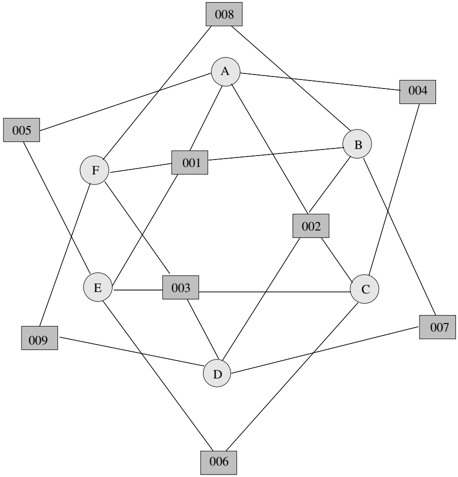
t_i	itemset			
001	A,	B,	E,	F
002	A,	B,	C,	D
003	A,	C,	E,	F
004	A,	C,		
005	A,	E,		
006	C,	E,		
007	B,	D,		
008	B,	F,		
009	D,	F,		

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

Symmetry Detection in Transaction Databases

- Convert the original problem $\mathcal{D}$ into a colored undirected graph $\mathcal{G}$, where vertices are labeled with colors.
- Look for the automorphism group of $\mathcal{G}$.
- Symmetries of $\mathcal{D}$ are equivalent to the automorphisms of the colored undirected graph $\mathcal{G}$ ([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		

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.

Outline

Symmetry-Based Pruning in Apriori-like algos

Apriori algorithm [Agrawal, 93]

→ Input: D Transaction dataset, λ: minimal support threshold
→ Proceeds by a level-wise search of the elements of $\mathcal{FIM}(\mathcal{D}, \lambda)$.

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

Symmetry-Based Pruning in Apriori-like algos

Algorithm 1: APRIORI

Data: D: Transaction dataset, λ: minimal support threshold
Result: the set of all frequent itemsets
$F_1 \leftarrow \{\text{frequent 1-itemsets}\}$;

Symmetry-Based Pruning in Apriori-like algos

Algorithm 2: Apriori

Data: D: Transaction dataset, λ: minimal support threshold

Result: the set of all frequent itemsets

$F_1 \leftarrow$ {frequent 1-itemsets};

for *($k = 2$; $F_{k-1} \neq \emptyset$; $k + +$)* **do**

  $F_k \leftarrow \emptyset$;

  $C_k \leftarrow CandidatesGen(F_{k-1})$;

Symmetry-Based Pruning in Apriori-like algos

Algorithm 3: APRIORI

Data: D: Transaction dataset, λ: minimal support threshold

Result: the set of all frequent itemsets

$F_1 \leftarrow$ {frequent 1-itemsets};

for $(k = 2; F_{k-1} \neq \emptyset; k++)$ **do**

  $F_k \leftarrow \emptyset$;

  $C_k \leftarrow CandidatesGen(F_{k-1})$;

  for $(c \in C_k)$ **do**

    $supp(c) \leftarrow SuppComp(c, \mathcal{D})$;

Symmetry-Based Pruning in Apriori-like algos

Algorithm 4: APRIORI

Data: D: Transaction dataset, λ: minimal support threshold
Result: the set of all frequent itemsets

$F_1 \leftarrow$ {frequent 1-itemsets};
for *($k = 2$; $F_{k-1} \neq \emptyset$; $k++$)* **do**
  $F_k \leftarrow \emptyset$;
  $C_k \leftarrow \mathit{CandidatesGen}(F_{k-1})$;
  for *($c \in C_k$)* **do**
    $\mathit{supp}(c) \leftarrow \mathit{SuppComp}(c, \mathcal{D})$;
    if *($\mathit{supp}(c) \geq \lambda$)* **then**
      $F_k \leftarrow F_k \cup \{c\}$;
return *($\bigcup_k F_k$)*;

Symmetry-Based Pruning in Apriori-like algos

Let $\mathcal{D}$ be a transaction database and λ a minimal support threshold. s.t.

- $\mathcal{I}(\mathcal{D}) = \{A, B, C, D\}$,
- $\sigma = (B, C)$ and $\sigma' = (A, C)(B, D)$ two symmetries of $\mathcal{D}$

Symmetry-Based Pruning in Apriori-like algos

Let $\mathcal{D}$ be a transaction database and λ a minimal support threshold. s.t.

- $\mathcal{I}(\mathcal{D}) = \{A, B, C, D\}$,
- $\sigma = (B, C)$ and $\sigma' = (A, C)(B, D)$ two symmetries of $\mathcal{D}$

Intermediary step

$\{A\}, \{B\}, \{C\}, \{D\} \in \mathcal{FIM}(\mathcal{D}, \lambda)$ and $\{A, B\} \notin \mathcal{FIM}(\mathcal{D}, \lambda)$

Symmetry-Based Pruning in Apriori-like algos

Let $\mathcal{D}$ be a transaction database and λ a minimal support threshold. s.t.

- $\mathcal{I}(\mathcal{D}) = \{A, B, C, D\}$,
- $\sigma = (B, C)$ and $\sigma' = (A, C)(B, D)$ two symmetries of $\mathcal{D}$

Intermediary step

$\{A\}, \{B\}, \{C\}, \{D\} \in \mathcal{FIM}(\mathcal{D}, \lambda)$ and $\{A, B\} \notin \mathcal{FIM}(\mathcal{D}, \lambda)$

anti-monotonicity $\rightarrow$ $\{A, B, C\}, \{A, B, D\}, \{A, B, C, D\} \notin \mathcal{FIM}(\mathcal{D}, \lambda)$

Symmetry-Based Pruning in Apriori-like algos

Let $\mathcal{D}$ be a transaction database and λ a minimal support threshold. s.t.

- $\mathcal{I}(\mathcal{D}) = \{A, B, C, D\}$,
- $\sigma = (B, C)$ and $\sigma' = (A, C)(B, D)$ two symmetries of $\mathcal{D}$

Intermediary step

$\{A\}, \{B\}, \{C\}, \{D\} \in \mathcal{FIM}(\mathcal{D}, \lambda)$ and $\{A, B\} \notin \mathcal{FIM}(\mathcal{D}, \lambda)$

anti-monotonicity $\rightarrow$ $\{A, B, C\}, \{A, B, D\}, \{A, B, C, D\} \notin \mathcal{FIM}(\mathcal{D}, \lambda)$

symmetries σ and σ' $\rightarrow$ $\{A, C\} \notin \mathcal{FIM}(\mathcal{D}, \lambda)$ and $\{C, D\} \notin \mathcal{FIM}(\mathcal{D}, \lambda)$

Symmetry-Based Pruning in Apriori-like algos

Let $\mathcal{D}$ be a transaction database and λ a minimal support threshold. s.t.

- $\mathcal{I}(\mathcal{D}) = \{A, B, C, D\}$,
- $\sigma = (B, C)$ and $\sigma' = (A, C)(B, D)$ two symmetries of $\mathcal{D}$

Intermediary step

$\{A\}, \{B\}, \{C\}, \{D\} \in \mathcal{FIM}(\mathcal{D}, \lambda)$ and $\{A, B\} \notin \mathcal{FIM}(\mathcal{D}, \lambda)$

anti-monotonicity	$\rightarrow$	$\{A, B, C\}, \{A, B, D\}, \{A, B, C, D\} \notin \mathcal{FIM}(\mathcal{D}, \lambda)$
symmetries σ and σ'	$\rightarrow$	$\{A, C\} \notin \mathcal{FIM}(\mathcal{D}, \lambda)$ and $\{C, D\} \notin \mathcal{FIM}(\mathcal{D}, \lambda)$
anti-monotonicity	$\rightarrow$	$\{A, C, D\}, \{B, C, D\} \notin \mathcal{FIM}(\mathcal{D}, \lambda)$.

Symmetry-Based Pruning in Apriori-like algos

Algorithm 5: $\textsc{Apriori}_{Sym}$

Data: D: database, λ: minimal support threshold, $\mathcal{S}$: symmetries in D

Result: the set of all frequent itemsets

$F_1 \leftarrow \{\text{frequent 1-itemsets}\}$;

Symmetry-Based Pruning in Apriori-like algos

Algorithm 6: APRIORI$_{Sym}$

Data: D: database, λ: minimal support threshold, $\mathcal{S}$: symmetries in D

Result: the set of all frequent itemsets

$F_1 \leftarrow$ {frequent 1-itemsets};

for $(k = 2;\ F_{k-1} \neq \emptyset;\ k++)$ **do**

  $F_k \leftarrow \emptyset$;

  $C_k \leftarrow CandidatesGen(F_{k-1})$;

Symmetry-Based Pruning in Apriori-like algos

Algorithm 7: $\textsc{Apriori}_{Sym}$

Data: D: database, λ: minimal support threshold, $\mathcal{S}$: symmetries in D

Result: the set of all frequent itemsets

$F_1 \leftarrow$ {frequent 1-itemsets};

for *($k = 2$; $F_{k-1} \neq \emptyset$; $k++$)* **do**

  $F_k \leftarrow \emptyset$;

  $C_k \leftarrow CandidatesGen(F_{k-1})$;

  for *($c \in C_k$)* **do**

    $supp(c) \leftarrow SuppComp(c, \mathcal{D})$;

Symmetry-Based Pruning in Apriori-like algos

Algorithm 8: APRIORI_{Sym}

Data: D: database, λ: minimal support threshold, $\mathcal{S}$: symmetries in D

Result: the set of all frequent itemsets

$F_1 \leftarrow$ {frequent 1-itemsets};

for *($k = 2$; $F_{k-1} \neq \emptyset$; $k + +$)* **do**

  $F_k \leftarrow \emptyset$;

  $C_k \leftarrow CandidatesGen(F_{k-1})$;

  for *($c \in C_k$)* **do**

    $supp(c) \leftarrow SuppComp(c, \mathcal{D})$;

    $S \leftarrow SymmGen(c, \mathcal{S})$;

Symmetry-Based Pruning in Apriori-like algos

Algorithm 9: $\textsc{Apriori}_{Sym}$

Data: D: database, λ: minimal support threshold, $\mathcal{S}$: symmetries in D

Result: the set of all frequent itemsets

$F_1 \leftarrow$ {frequent 1-itemsets};
for *($k = 2$; $F_{k-1} \neq \emptyset$; $k++$)* **do**
    $F_k \leftarrow \emptyset$;
    $C_k \leftarrow CandidatesGen(F_{k-1})$;
    for *($c \in C_k$)* **do**
        $supp(c) \leftarrow SuppComp(c, \mathcal{D})$;
        $S \leftarrow SymmGen(c, \mathcal{S})$;
        if *($supp(c) \geq \lambda$)* **then**
            $F_k \leftarrow F_k \cup \{c\} \cup S$;
        $C_k = C_k \setminus S$;
return *($\bigcup_k F_k$)*;

Symmetry-Based Pruning in Apriori-like algos: Example 2

- 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.

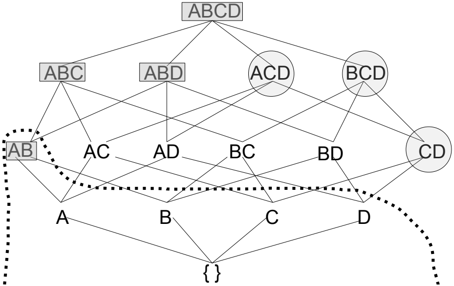


Figure : Symmetry Pruning

Outline

Experimental results

dataset	#trans	#items	density
Zoo	101	43	39%
Mushroom	8 142	117	18%
Australian	690	55	25%
Solar flare	323	40	32%
Letter-recognition	20 000	74	23%
BMS-WebView-2	77 512	3 341	0.14%

Table : Description of the datasets

Experimental results

dataset	#trans	#items	density
Zoo	101	43	39%
Mushroom	8 142	117	18%
Australian	690	55	25%
Solar flare	323	40	32%
Letter-recognition	20 000	74	23%
BMS-WebView-2	77 512	3 341	0.14%

Table : Description of the datasets

dataset	#sym	time (s)
Zoo	2	0.02
Mushroom	10	0.94
Australian	2	0.06
Solar flare	2	0.05
Letter-recognition	0	2.12
BMS-WebView-2	10	4.31

Table : Symmetry Extraction Time

Experimental results

Dataset	*minfr*	Sym Pruning	#db scans	$\mathcal{M}_{SR}$	#freq	#freq$_{Sym}$	#infreq$_{Sym}$	$\mathcal{M}_{\mathcal{TA}}$
Australian	1%	−−	479 402	24%	426 763	−−	−−	23%
		✓	364 822			100 961	13 613	
	5%	−−	28 535	22%	20 386	−−	−−	22%
		✓	22 288			4.461	1 886	
Solar-Flare	1%	−−	147 270	9%	145 893	−−	−−	8%
		✓	133 056			14 128	1 291	
	15%	−−	5 684	8%	5 495	−−	−−	9%
		✓	5 203			448	33	
Mushroom	5%	−−	3 764 532	0.4%	3 755 511	−−	−−	6%
		✓	3 748 084			16 384	8 957	
Zoo	5%	−−	587 782	20%	486 099	−−	−−	15%
		✓	487 224			100 480	78	
	15%	−−	103 318	23%	102 440	−−	−−	17%
		✓	79 492			23 776	50	
Letter-recognition	5%	−−	32 680	0%	15 719	−−	−−	0%
		✓	32 680			0	0	
BMS-WebView-2	1%	−−	4 908	0%	81	−−	−−	0%
		✓	4 908			0	0	

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.