Frequent Pattern Mining

Toon Calders
University of Antwerp
Summary

- Frequent Itemset Mining
- Algorithms
- Constraint Based Mining
- Condensed Representations
Frequent Itemset Mining

- Market-Basket Analysis

<table>
<thead>
<tr>
<th>TID</th>
<th>Item 0</th>
<th>Item 1</th>
<th>Item 2</th>
<th>Item 3</th>
<th>Item 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

transaction identifier

items

transaction

- Frequent Itemset RMining
- Market-Basket Analysis

transaction items

transaction identifier
Frequent Itemset Mining

- support(I): number of transactions “containing I”

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Support(BC) = 3
Support(ACD) = 2
Frequent Itemset Mining Problem

Given D, minsup
Find all sets I with support(I) ≥ minsup

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

minsup=2

{}, A, B, C, D,
AC, AD, BC, BD, CD, ACD
Why?

- Important component in mining algorithms
- Sufficient statistics for interestingness measures
  - Confidence $X \rightarrow Y : \text{Support}(XY)/\text{Support}(X)$
  - Contingency tables (correlation, $X^2$)

<table>
<thead>
<tr>
<th></th>
<th>$X$</th>
<th>$\neg X$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y$</td>
<td>$s(XY)$</td>
<td>$s(Y) - s(XY)$</td>
</tr>
<tr>
<td>$\neg Y$</td>
<td>$s(X) - s(XY)$</td>
<td>$s({}) - s(X) - s(Y) + s(XY)$</td>
</tr>
</tbody>
</table>
Summary

- Frequent Itemset Mining
- Algorithms
- Constraint Based Mining
- Condensed Representations
Algorithms

There exist hundreds of algorithms that solve FIM (or related problems)

- AIS, Apriori, AprioriTID, AprioriHybrid, FPGrowth, FPGrowth*, Eclat, dEclat, Pincer-search, ABS, DCI, kDCI, LCM, AIM, PIE, ARMOR, AFOPT, COFI, Patricia, MAXMINER, MAFIA, NDI-ALL, …
Algorithms

- There exist hundreds of algorithms that solve FIM (or related problems)
- Concentrate on the most important pruning principle:
  - Monotonicity
and the two main search strategies:
  - Breadth-first
  - Depth-first
Monotonicity Principle

- If $I \subseteq J$, then $\text{support}(I) \geq \text{support}(J)$
- Therefore, if $I$ is infrequent, then all its supersets are infrequent as well.

- All FIM algorithms rely heavily on this principle to prune large parts of the search space.
Search Space

AB infrequent

A B C D

AC AD BC BD CD

ACD BCD

A C B D
Levelwise Algorithm

- Exploits monotonicity as much as possible.
- Search Space is traversed bottom-up, level by level
- Support of an itemset is only counted in the database if all its subsets were frequent.
Apriori

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\text{minsup}=2
\]

Candidates

\[
\emptyset
\]
<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Apriori

\[
\text{minsup} = 2
\]
$$\text{minsup}=2$$

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Apriori
Apriori

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ \text{minsup} = 2 \]

A \rightarrow B \rightarrow C \rightarrow D
Apriori

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\( \text{mins} \text{sup}=2 \)

```
11110
11011
00110
12345
```
Apriori

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ \text{minsup} = 2 \]
Apriori

minsup=2

candidates

A 2
B 4
C 4
D 3

{}
Apriori

TID | A | B | C | D
---|---|---|---|---
1  | 0 | 1 | 1 | 0
2  | 0 | 1 | 1 | 0
3  | 1 | 0 | 1 | 1
4  | 1 | 1 | 1 | 1
5  | 0 | 1 | 0 | 1

\text{minsup}=2
Apriori

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

\[\text{minsups} = 2\]

Candidates

\[\{\}\]
Apriori

minsуп=2
Depth-First Algorithms

Find all frequent itemsets

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Find all frequent itemsets, with D

Find all frequent itemsets, without D

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Depth-First Algorithm

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

- **TID**: Transaction Identification Number
- **A, B, C, D**: Attributes
- **DB[D]**: Database on attribute D
- **DB[BD]**: Database on attributes B and D
- **DB[CD]**: Database on attributes C and D
- **DB[BC]**: Database on attributes B and C
- **ACD**: Association of attributes A, C, D
- **AD, BD, CD**: Association of attributes A and D, B and D, C and D
- **AC, BC**: Association of attributes A and C, B and C

Additional tables:

- **DB[C]**: Database on attribute C
  - | TID | A | B |
  - | 1   | 0 | 1 |
  - | 2   | 0 | 1 |
  - | 3   | 1 | 0 |
  - | 4   | 1 | 1 |

- **DB[B]**: Database on attribute B
  - | TID | A |
  - | 1   | 1 |
  - | 2   | 1 |
  - | 3   | 1 |
  - | 4   | 1 |
Breadth-First vs Depth-First

- Depth-first outperforms breadth-first
  - Number of frequent itemsets is very high
  - Database is relatively small
- Breadth-first outperforms depth-first
  - Number of frequent sets is small
  - Database is large
- Differences usually very small
Summary

- Frequent Itemset Mining
- Algorithms
- Constraint Based Mining
- Condensed Representations
Mining With Constraints

- Reduce output size, user sets focus
  - itemsets of size > 5
  - sets of products with cost less than 10 EUR
  - sets that contain A, B, or C.
  - sets that are frequent in dataset $D_1$, but infrequent in $D_2$
Mining With Constraints

- Types of constraints
  - (Anti-)Monotone,
  - Succinct
  -Convertible

- Two Approaches
  - Pushing constraints into the mining algorithm
  - Changing the Database
Types of Constraints

- Anti-monotone
  - Support, size < 10, …
Types of Constraints

- Monotone
  - Cost >10EUR, Contains A, B, or C, …
Types of Constraints

- **Succinct**
  - Can be expressed using minus and union on a fixed number of powersets
    - E.g., Contains A or B, but not C: $2^I - C \cap 2^I - AB$
  - Can be generated efficiently

- **Convertible anti-monotone**
  - Anti-Monotone w.r.t. prefix-order
    - E.g. $\text{avg}(I.\text{price}) < 10$ EUR when ordered ascending by price.
Mining With Constraints

Two approaches:

- Pushing constraints deep in data mining algorithm
- Changing database such that
  - Support of itemsets satisfying the constraint does not change
  - The support of itemsets that do not satisfy the constraint decreases
Pushing Constraints

Monotone

Anti-monotone

Frequency
Pushing Constraints

- Trade-off
  - Pushing monotone constraints
  - vs. anti-monotone pruning

- Not always better to push monotone constraints
  - E.g. Size > 10 …
Changing the Database

- ExAnte Algorithm
  - Exploit Monotone and Anti-monotone constraints
  - A transaction that does not satisfy a monotone constraint will not contribute to any itemset satisfying the constraints
    - E.g. constraint “size > 10”: every transaction of size < 10 can be thrown away!
Changing the Database

\( \text{minsup} = 3 \quad \text{anti-mon.} \)

\( \text{size} \geq 4 \quad \text{monotone} \)

<table>
<thead>
<tr>
<th>ID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Summary

- Frequent Itemset Mining
- Algorithms
- Constraint Based Mining
- Condensed Representations
Condensed Representations

- Sometimes, the output of frequent set mining remains too large:
  - Huge number of items
  - Highly correlated
  - High support items

- Hence, instead of mining all itemsets
  - Condensed representation
Condensed Representations

- Closed sets
  - Divide frequent itemsets into equivalence classes
  - Two itemsets are equivalent if they occur in the same transactions
  - Closed set: maximal element in an equivalence class
Closed Itemsets

- All sets in the same equivalence class have the same support
  - Occur in the same transactions
- Maximal element in an equivalence class is unique
  - If two itemsets occur in the same transactions, then so does their union
Closed Itemsets

<table>
<thead>
<tr>
<th>TID</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Diagram showing the closed itemsets with nodes representing items A, B, C, D and edges connecting itemsets.
Closed Itemsets

- Has nice mathematical properties
  - Closed sets form a lattice
  - Galois connection
- Efficient algorithms to find them
- Based on the closed sets, it is easy to find the support of the other itemsets.
Closed Itemsets

- Interesting class of patterns
  - Maximal frequent itemsets are closed sets
  - Highest correlation between items
  - Strongest association rules
- Significant reduction of number of itemsets
  - Especially with small number of large transactions
Non-Derivable Itemsets

- Based on redundancies
  - How do supports interact?

- What information about unknown supports can we derive from known supports?
  - Concise representation: only store relevant part of the supports
Redundancies

- Agrawal et al. (Monotonicity)
  - $\text{Supp}(AX) \leq \text{Supp}(A)$

- Boullicaut et al., Lakhal et al. (Free sets)
  - If $\text{Supp}(A) = \text{Supp}(AB)$ (Closed sets)
  - Then $\text{Supp}(AX) = \text{Supp}(AXB)$
Redundancies

- **Bayardo**
  **(MAXMINER)**
  \[ \text{Supp}(ABX) \geq \text{Supp}(AX) - (\text{Supp}(X) - \text{Supp}(BX)) \]
  drop \((X, B)\)

- **Bykowski, Rigotti**
  **(Disjunction-free sets)**
  if \(\text{Supp}(ABC) = \text{Supp}(AB) + \text{Supp}(AC) - \text{Supp}(A)\), then \(\text{Supp}(ABCX)\) can be derived from \(ABX, ACX, AX\)
The Inclusion – Exclusion Principle

\[ |A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C| \]
Deduction Rules via Inclusion-Exclusion

- Let A, B, C, … be items
- Let A’ correspond with the set
  \{ \text{transaction } t \mid t \text{ contains } A \} 
- \( AB' = A' \cap B' \)

Then: \( \text{Supp}(ABC) = |ABC'| \)
Deduction Rules via Inclusion-Exclusion

Inclusion-exclusion principle:
\[ | A' \cup B' \cup C' | = |A'| + |B'| + |C'| \]
\[- |AB'| - |AC'| - |BC'| \]
\[+ |ABC'| \]

Thus, since \(| A' \cup B' \cup C' | \leq n,\)

\[
\text{Supp}(ABC) \leq s(AB) + s(AC) + s(BC) \\
- s(A) - s(B) - s(C) + n
\]
Complete Set for Supp(ABC)

0\[s_{ABC} \geq 0\]

1
\[s_{ABC} \leq s_{AB}\]
\[s_{ABC} \leq s_{AC}\]
\[s_{ABC} \leq s_{BC}\]

Monotonicity

2
\[s_{ABC} \geq s_{AB} + s_{AC} - s_A\]
\[s_{ABC} \geq s_{AB} + s_{BC} - s_B\]
\[s_{ABC} \geq s_{AC} + s_{BC} - s_C\]

Disjunction-Free

3
\[s_{ABC} \leq s_{AB} + s_{AC} + s_{BC} - s_A - s_B - s_C + n\]

Free, Closed
Derivable Itemsets

Given: $\text{Supp}(I)$ for all $I \subset J$
Lower bound on $\text{Supp}(J) = l$
Upper bound on $\text{Supp}(J) = u$

- Without counting: $\text{Supp}(J) \in [l,u]$
- $J$ is a **derivable itemset** (DI) iff $l = u$
  - We know $\text{Supp}(J)$ **exactly** without counting!
Derivable Itemsets

J derivable itemset:
- No need to \textbf{count} \text{Supp}(J)
- No need to \textbf{store} \text{Supp}(J)
  - We can use the deduction rules

Concise representation:
\[ C = \{ ( J, \text{Supp}(J) ) | J \text{ not derivable from } \text{Supp}(I), I \subset J \} \]
Derivable Itemsets

Theorem (Monotonicity)
If $J \subseteq K$, $J$ derivable, then $K$ derivable.

Moreover:
The width of the interval for $J \cup \{A\}$ is at most half the size of the interval for $J$. 
IV. Evaluation --- Theoretical

- Interval widths decrease exponentially
  - Half each step

- Non-derivable itemset can never be larger than \( \log(|\text{Database}|) \)
  - Independent of sparse, dense, ...
Evaluation --- Empirically

- Size NDI vs. frequent itemsets
- Comparison with Other Concise Reps
Evaluation

- Number of frequent NDI s considerable smaller than number of frequent itemsets

- Algorithm is efficient
  - Calculating NDI + deducing DIs often outperforms Apriori
Condensed Representations

- Many other representations
  - Free sets
  - Disjunction-free sets
  - Generalized disjunction-free sets
  - ...

- Closed sets and NDIs provable the smallest ones
Conclusion

- Depth-first vs Breadth-first algorithms for FIM
- Constraint mining to incorporate user focus
  - Pushing constraints vs changing database
- Condensed Representations
  - Closed sets
  - Non-Derivable Itemsets
Topics Not Covered …

Parallel algorithms for FIM
Incremental FIM
Generalized, Quantitative, Multi-level, Fuzzy ARs
Coupling FIM with RDBMS
Privacy Preserving ARM
Computational Complexity Results
  Inverse mining problem
Emerging Patterns, jumping emerging patterns
  Dependency value, $X^2$
  Lift, gain
  Block support, tilings,
  …