Association Rules
Salvatore Ruggieri
Computer Science Department, University of Pisa
Let $I = \{ a_1, \ldots, a_n \}$ be a finite set
- $a_i \in I$ is called an item

A itemset $I$ is a subset of $I$
- $I \subseteq I$

A transaction $t$ is an itemset with an identifier
- $t = (i, I)$ with $I \subseteq I$ also written $t_i \subseteq I$

A transaction database is a finite set of transactions
- $D = \{ t_i \mid i = 1 \ldots d, t_i \subseteq I \}$
Format of transaction db

- Transactional
  - A row for each transaction (id not necessary)
  - List of items in the transaction

<table>
<thead>
<tr>
<th>milk, sugar, water</th>
</tr>
</thead>
<tbody>
<tr>
<td>beer, diapers</td>
</tr>
<tr>
<td>...</td>
</tr>
</tbody>
</table>

- Not available in Weka
Format of transaction db

- Tabular
  - Two columns
    - transaction ID
    - item

<table>
<thead>
<tr>
<th>tID</th>
<th>item</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>milk</td>
</tr>
<tr>
<td>1</td>
<td>sugar</td>
</tr>
<tr>
<td>1</td>
<td>water</td>
</tr>
<tr>
<td>2</td>
<td>beer</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

Filter available in Weka: [denormalize](#)

- Use the GUI Chooser -> Tools -> Package Manager to install it
Format of transaction db

- **Binary**
  - A column for each item
  - A row for each transaction (id not necessary)
- **Cell value**
  - true (false) if the item is (not) in the transaction

<table>
<thead>
<tr>
<th>milk</th>
<th>sugar</th>
<th>…</th>
<th>diapers</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>true</td>
<td>…</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>false</td>
<td>…</td>
<td>true</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

- In Weka use ‘?’ for false
Format of transaction db

- Relational
  - Item are of the form `att=value`
    - `att` is an attribute, `value` is a value in its domain
  - A row for each transaction (id not necessary)
- Cell value
  - `value` if the item `att=value` is in the transaction

<table>
<thead>
<tr>
<th>income</th>
<th>status</th>
<th>...</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>high</td>
<td>married</td>
<td></td>
<td>20-30</td>
</tr>
<tr>
<td>medium</td>
<td>single</td>
<td></td>
<td>40-50</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Available in Weka
Support & Confidence

- **Association rule** $X \rightarrow Y$
  - $X, Y$ itemsets and $X \cap Y = \emptyset$

- **Classification rule** $X \rightarrow C$
  - $X$ itemset, $C$ class item, $C \notin X$
  - Common in the relational format

- **Support of an itemset**
  - $\text{supp}(l) = |\{ t \in D \mid l \subseteq t \}|$
  - relative support: $\text{supp}(l)/|D|$
Support & Confidence

- 4-fold contingency table

<table>
<thead>
<tr>
<th></th>
<th>Y</th>
<th>¬Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>¬X</td>
<td>c</td>
<td>d</td>
</tr>
</tbody>
</table>

- \(\text{supp}(X \rightarrow Y) = a = \text{supp}(X,Y)\)
- \(\text{conf}(X \rightarrow Y) = a/(a+b) = \text{supp}(X,Y)/\text{supp}(X)\)
- \(\text{coverage}(X \rightarrow Y) = a+b = \text{supp}(X)\)
- \(\text{lift}(X \rightarrow Y) = \text{conf}(X \rightarrow Y)/\text{conf}(true \rightarrow Y) = \frac{\text{supp}(X,Y)}{(\text{supp}(X) \times \text{supp}(Y))}\)
Software for AR mining

- Weka
  - Binary and relational format

- Frida
  - [http://www.borgelt.net/frida.html](http://www.borgelt.net/frida.html)
  - Transactional format

- SQL Server Analysis Services
  - Relational format

- A lot of research and commercial systems
Demo and practice

- Demo on the supermarket dataset
- Practice on the German credit dataset

**Objective:**
- Find conditions of past bad credit

**Method**
- Find classification rules with `class=bad`
- Rank them wrt which measure?