Outline

1. What is Association Analysis?
2. Frequent Itemset Generation
3. Rule Generation
4. Interestingness Measures
5. Handling Continuous and Categorical Attributes
Association Analysis

• First algorithms developed in the early 90s at IBM by Agrawal & Srikant
• Motivation
  – Availability of barcode cash registers
Association Analysis

• initially used for Market Basket Analysis
  – to find how items purchased by customers are related
• later extended to more complex data structures
  – sequential patterns (see Data Mining II)
  – subgraph patterns
• and other application domains
  – life science
  – social science
  – web usage mining
Simple Approaches

• To find out if two items x and y are bought together, we can compute their correlation

• E.g., Pearson's correlation coefficient:

\[
\frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2} \sqrt{\sum (y_i - \bar{y})^2}}
\]

• Numerical coding:
  – 1: item was bought
  – 0: item was not bought

• $\bar{x}$: average of x (i.e., how often x was bought)
## Correlation Analysis in RapidMiner

![Correlation Matrix](attachment:image.png)

<table>
<thead>
<tr>
<th>Attributes</th>
<th>ThinkPad X2</th>
<th>Asus EeePC</th>
<th>HP Laserjet</th>
<th>2 GB DDR3</th>
<th>8 GB DDR3</th>
<th>Lenovo Tab</th>
<th>Netbook-Sc</th>
<th>HP CE50</th>
<th>LT Laser M.</th>
<th>LT Minimaus</th>
</tr>
</thead>
<tbody>
<tr>
<td>ThinkPad X2</td>
<td>1</td>
<td>-1</td>
<td>0.356</td>
<td>-0.816</td>
<td>0.612</td>
<td>0.583</td>
<td>-0.667</td>
<td>0.356</td>
<td>0.167</td>
<td>-0.408</td>
</tr>
<tr>
<td>Asus EeePC</td>
<td>-1</td>
<td>1</td>
<td>-0.356</td>
<td>0.816</td>
<td>-0.612</td>
<td>-0.583</td>
<td>0.667</td>
<td>-0.356</td>
<td>-0.167</td>
<td>0.408</td>
</tr>
<tr>
<td>HP Laserjet</td>
<td>0.356</td>
<td>-0.356</td>
<td>1</td>
<td>-0.218</td>
<td>0.356</td>
<td>0.535</td>
<td>1</td>
<td>-0.089</td>
<td>-0.655</td>
<td>0.200</td>
</tr>
<tr>
<td>2 GB DDR3</td>
<td>-0.816</td>
<td>0.816</td>
<td>-0.218</td>
<td>1</td>
<td>-0.500</td>
<td>-0.816</td>
<td>0.816</td>
<td>-0.218</td>
<td>0</td>
<td>0.200</td>
</tr>
<tr>
<td>8 GB DDR3</td>
<td>0.612</td>
<td>-0.612</td>
<td>-0.327</td>
<td>-0.500</td>
<td>1</td>
<td>0.102</td>
<td>-0.408</td>
<td>-0.327</td>
<td>0.102</td>
<td>0</td>
</tr>
<tr>
<td>Lenovo Tab</td>
<td>0.583</td>
<td>-0.583</td>
<td>0.356</td>
<td>-0.816</td>
<td>1</td>
<td>0.102</td>
<td>0.356</td>
<td>0.167</td>
<td>0.408</td>
<td>0</td>
</tr>
<tr>
<td>Netbook-Sc</td>
<td>-0.667</td>
<td>0.667</td>
<td>-0.535</td>
<td>0.816</td>
<td>-0.408</td>
<td>-0.667</td>
<td>1</td>
<td>-0.535</td>
<td>0.167</td>
<td>0.408</td>
</tr>
<tr>
<td>HP CE50</td>
<td>0.356</td>
<td>-0.356</td>
<td>1</td>
<td>-0.218</td>
<td>0.356</td>
<td>0.535</td>
<td>1</td>
<td>-0.089</td>
<td>-0.655</td>
<td>0.200</td>
</tr>
<tr>
<td>LT Laser M.</td>
<td>0.167</td>
<td>-0.167</td>
<td>-0.089</td>
<td>0</td>
<td>0.102</td>
<td>-0.250</td>
<td>0.167</td>
<td>-0.089</td>
<td>1</td>
<td>-0.408</td>
</tr>
<tr>
<td>LT Minimaus</td>
<td>-0.408</td>
<td>0.408</td>
<td>-0.655</td>
<td>0.200</td>
<td>0</td>
<td>0</td>
<td>0.408</td>
<td>-0.655</td>
<td>-0.408</td>
<td>1</td>
</tr>
</tbody>
</table>
Correlation vs. Causality

http://xkcd.com/552/
Association Analysis

Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction.

Market-Basket transactions

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bread, Milk</td>
</tr>
<tr>
<td>2</td>
<td>Bread, Diaper, Beer, Eggs</td>
</tr>
<tr>
<td>3</td>
<td>Milk, Diaper, Beer, Coke</td>
</tr>
<tr>
<td>4</td>
<td>Bread, Milk, Diaper, Beer</td>
</tr>
<tr>
<td>5</td>
<td>Bread, Milk, Diaper, Coke</td>
</tr>
</tbody>
</table>

Examples of Association Rules

{Diaper} → {Beer},
{Milk, Bread} → {Eggs, Coke},
{Beer, Bread} → {Milk},

Implication means co-occurrence, not causality!
Definition: Frequent Itemset

- **Itemset**
  - A collection of one or more items
    - Example: \{Milk, Bread, Diaper\}
  - k-itemset
    - An itemset that contains k items

- **Support count ($\sigma$)**
  - Frequency of occurrence of an itemset
  - E.g. $\sigma(\{Milk, Bread, Diaper\}) = 2$

- **Support**
  - Fraction of transactions that contain an itemset
  - E.g. $s(\{Milk, Bread, Diaper\}) = 2/5$

- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a $\minsup$ threshold.
Definition: Association Rule

- **Association Rule**
  - An implication expression of the form $X \rightarrow Y$, where $X$ and $Y$ are itemsets.
  - An association rule states that when $X$ occurs, $Y$ occurs with certain probability.

- **Rule Evaluation Metrics**
  - **Support** $(s)$
    Fraction of transactions that contain both $X$ and $Y$.
  - **Confidence** $(c)$
    Measures how often items in $Y$ appear in transactions that contain $X$.

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bread, Milk</td>
</tr>
<tr>
<td>2</td>
<td>Bread, Diaper, Beer, Eggs</td>
</tr>
<tr>
<td>3</td>
<td>Milk, Diaper, Beer, Coke</td>
</tr>
<tr>
<td>4</td>
<td>Bread, Milk, Diaper, Beer</td>
</tr>
<tr>
<td>5</td>
<td>Bread, Milk, Diaper, Coke</td>
</tr>
</tbody>
</table>

\[
s(X \rightarrow Y) = \frac{|X \cup Y|}{|T|}
\]

\[
c(X \rightarrow Y) = \frac{s(X \cup Y)}{s(X)}
\]
Definition: Association Rule

- **Association Rule**
  - An implication expression of the form \( X \rightarrow Y \), where \( X \) and \( Y \) are itemsets
  - An association rule states **that** when \( X \) occurs, \( Y \) occurs with certain **probability**.

- **Rule Evaluation Metrics**
  - **Support** \( (s) \)
    - Fraction of transactions that contain both \( X \) and \( Y \)
  - **Confidence** \( (c) \)
    - Measures how often items in \( Y \) appear in transactions that contain \( X \)

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bread, Milk</td>
</tr>
<tr>
<td>2</td>
<td>Bread, Diaper, Beer, Eggs</td>
</tr>
<tr>
<td>3</td>
<td>Milk, Diaper, Beer, Coke</td>
</tr>
<tr>
<td>4</td>
<td>Bread, Milk, Diaper, Beer</td>
</tr>
<tr>
<td>5</td>
<td>Bread, Milk, Diaper, Coke</td>
</tr>
</tbody>
</table>

**Example:**

\[ \{ \text{Milk, Diaper} \} \Rightarrow \text{Beer} \]

\[
s = \frac{\sigma(\text{Milk, Diaper, Beer})}{|T|} = \frac{2}{5} = 0.4
\]

\[
c = \frac{\sigma(\text{Milk, Diaper, Beer})}{\sigma(\text{Milk, Diaper})} = \frac{2}{3} = 0.67
\]
Association Rule Mining Task

- Given a set of transactions $T$, the goal of association rule mining is to find all rules having
  - support $\geq \text{minsup}$ threshold
  - confidence $\geq \text{minconf}$ threshold

- $\text{minsup}$ and $\text{minconf}$ are provided by the user

- Brute-force approach:
  1. List all possible association rules
  2. Compute the support and confidence for each rule
  3. Remove rules that fail the $\text{minsup}$ and $\text{minconf}$ thresholds

  $\Rightarrow$ Computationally prohibitive due to large number of candidates!
## Mining Association Rules

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bread, Milk</td>
</tr>
<tr>
<td>2</td>
<td>Bread, Diaper, l</td>
</tr>
<tr>
<td>3</td>
<td>Milk, Diaper, Beer</td>
</tr>
</tbody>
</table>

**Example of Rules:**

\[
\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\} \ (s=0.4, \ c=0.67) \\
\{\text{Milk, Beer}\} \rightarrow \{\text{Diaper}\} \ (s=0.4, \ c=1.0) \\
\{\text{Diaper, Beer}\} \rightarrow \{\text{Milk}\} \ (s=0.4, \ c=0.67) \\
\{\text{Beer}\} \rightarrow \{\text{Milk, Diaper}\} \ (s=0.4, \ c=0.67) \\
\{\text{Diaper}\} \rightarrow \{\text{Milk, Beer}\} \ (s=0.4, \ c=0.5) \\
\{\text{Milk}\} \rightarrow \{\text{Diaper, Beer}\} \ (s=0.4, \ c=0.5)
\]

**Observations:**

- All the above rules are binary partitions of the same itemset: \{\text{Milk, Diaper, Beer}\}
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may decouple the support and confidence requirements

\[
s(X \rightarrow Y) = \frac{|X \cup Y|}{|T|}
\]
Apriori Algorithm: Basic Idea

• Two-step approach

• First: Frequent Itemset Generation
  – Generate all itemsets whose support ≥ minsup

• Second: Rule Generation
  – Generate high confidence rules from each frequent itemset
  – where each rule is a binary partitioning of a frequent itemset

• However: Frequent itemset generation is still computationally expensive....
Frequent Itemset Generation

Given \( d \) items, there are \( 2^d \) candidate itemsets!
Brute-force Approach

• Example:
  – Amazon has 10 million books (i.e., Amazon Germany, as of 2011)
• That is \(2^{10000000}\) possible itemsets
• As a number:
  – \(9.04981... \times 10^{3010299}\)
  – That is: a number with 3 million digits!

• However:
  – most itemsets will not be important at all
  – e.g., books on Chinese calligraphy and data mining bought together
  – thus, smarter algorithms should be possible
Brute-force Approach

- Each itemset in the lattice is a candidate frequent itemset
- Count the support of each candidate by scanning the database
- Match each transaction against every candidate

Complexity \( \sim O(NMw) \) \( \rightarrow \) Expensive since \( M = 2^d \)

A smarter algorithm is required
Reducing the Number of Candidates

- **Apriori Principle**

  If an itemset is frequent, then all of its subsets must also be frequent.

- Apriori principle holds due to the following property of the support measure:

  \[ \forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y) \]

  - Support of an itemset never exceeds the support of its subsets
  - This is known as the **anti-monotone** property of support
Illustrating the Apriori Principle

Found to be Infrequent

Pruned supersets
Illustrating the Apriori Principle

<table>
<thead>
<tr>
<th>Item</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bread</td>
<td>4</td>
</tr>
<tr>
<td>Coke</td>
<td>2</td>
</tr>
<tr>
<td>Milk</td>
<td>4</td>
</tr>
<tr>
<td>Beer</td>
<td>3</td>
</tr>
<tr>
<td>Diaper</td>
<td>4</td>
</tr>
<tr>
<td>Eggs</td>
<td>1</td>
</tr>
</tbody>
</table>

No need to generate candidates involving Coke or Eggs.

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Bread, Milk}</td>
<td>3</td>
</tr>
<tr>
<td>{Bread, Beer}</td>
<td>2</td>
</tr>
<tr>
<td>{Bread, Diaper}</td>
<td>3</td>
</tr>
<tr>
<td>{Milk, Beer}</td>
<td>2</td>
</tr>
<tr>
<td>{Milk, Diaper}</td>
<td>3</td>
</tr>
<tr>
<td>{Beer, Diaper}</td>
<td>3</td>
</tr>
</tbody>
</table>

No need to generate candidate {Milk, Diaper, Beer}

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Bread, Milk, Diaper}</td>
<td>3</td>
</tr>
</tbody>
</table>

Minimum Support = 3
The Apriori Algorithm

1. Let k=1
2. Generate frequent itemsets of length 1
3. Repeat until no new frequent itemsets are identified
   1. Generate length (k+1) candidate itemsets from length k frequent itemsets; increase k
   2. Prune candidate itemsets that cannot be frequent because they contain subsets of length k that are infrequent (Apriori Principle)
   3. Count the support of each remaining candidate by scanning the DB
   4. Eliminate candidates that are infrequent, leaving only those that are frequent
Example: Finding frequent itemsets

```
minsup=0.5  Dataset T

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>T100</td>
<td>1, 3, 4</td>
</tr>
<tr>
<td>T200</td>
<td>2, 3, 5</td>
</tr>
<tr>
<td>T300</td>
<td>1, 2, 3, 5</td>
</tr>
<tr>
<td>T400</td>
<td>2, 5</td>
</tr>
</tbody>
</table>
```

1. scan T
   → Cand1: {1}:2, {2}:3, {3}:3, {4}:1, {5}:3
   → Freq1: {1}:2, {2}:3, {3}:3, {5}:3
   → Cand2: {1,2}, {1,3}, {1,5}, {2,3}, {2,5}, {3,5}

2. scan T
   → Cand2: {1,2}:1, {1,3}:2, {1,5}:1, {2,3}:2, {2,5}:3, {3,5}:2
   → Freq2: {1,3}:2, {2,3}:2, {2,5}:3, {3,5}:2
   → Cand3: {1,3}, {2,3}, {2,5}, {3,5}

3. scan T
   → Cand3: {2, 3, 5}:2
   → Freq3: {2, 3, 5}
```
Rule Generation

- Given a frequent itemset \( L \), find all non-empty subsets \( f \subset L \) such that \( f \rightarrow L - f \) satisfies the minimum confidence requirement.

Example Frequent Itemset:
\[ \{ \text{Milk, Diaper, Beer} \} \]

Example Rule:
\[ \{ \text{Milk, Diaper} \} \rightarrow \text{Beer} \]

\[
c = \frac{\sigma(\text{Milk, Diaper, Beer})}{\sigma(\text{Milk, Diaper})} = \frac{2}{3} = 0.67
\]
Challenge: Large Number of Rules

If \{A,B,C,D\} is a frequent itemset, then the candidate rules are:

- $ABC \rightarrow D$, $ABD \rightarrow C$, $ACD \rightarrow B$, $BCD \rightarrow A$,
- $A \rightarrow BCD$, $B \rightarrow ACD$, $C \rightarrow ABD$, $D \rightarrow ABC$,
- $AB \rightarrow CD$, $AC \rightarrow BD$, $AD \rightarrow BC$, $BC \rightarrow AD$,
- $BD \rightarrow AC$, $CD \rightarrow AB$,

If $|L| = k$, then there are $2^k - 2$ candidate association rules (ignoring $L \rightarrow \emptyset$ and $\emptyset \rightarrow L$).
Rule Generation

- **How to efficiently generate rules from frequent itemsets?**
  - In general, confidence does not have an anti-monotone property
    \[ c(ABC \rightarrow D) \text{ can be larger or smaller than } c(AB \rightarrow D) \]
  - But confidence of rules generated from the same itemset has an anti-monotone property
  - e.g., \( L = \{A,B,C,D\} \):
    \[ c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD) \]
  - Confidence is **anti-monotone w.r.t. number of items on the RHS of the rule**
Rule Generation

• Confidence is anti-monotone w.r.t. number of items on the RHS of the rule
  – i.e., “moving elements from left to right” cannot increase confidence
  – reason:

\[
\begin{align*}
  c(AB \rightarrow C) & := \frac{s(ABC)}{s(AB)} & c(A \rightarrow BC) & := \frac{s(ABC)}{s(A)} \\
  c(A \rightarrow BC) & := \frac{s(ABC)}{s(A)}
\end{align*}
\]

– Due to anti-monotone property of support, we know
  • \( S(AB) \leq S(A) \)
  – Hence
    • \( c(AB \rightarrow C) \geq C(A \rightarrow BC) \)
Rule Generation for Apriori Algorithm

- Candidate rule is generated by merging two rules that share the same prefix in the rule consequent

- \( \text{join}(CD=>AB, BD=>AC) \)
  - would produce the candidate rule \( D=>ABC \)

- Prune rule \( D=>ABC \)
  - if its subset \( AD=>BC \) does not have high confidence

- All the required information for confidence computation has already been recorded during itemset generation.
  \[ c(X \rightarrow Y) := \frac{s(X \cup Y)}{s(X)} \]
  → No need to see the data anymore!
Complexity of Apriori Algorithm

• Expensive part is scanning the database
  – i.e., when counting the support of frequent itemsets
• The database is scanned once per pass of frequent itemset generation
  – one pass to count frequencies of 1-itemsets
  – one pass to count frequencies of 2-itemsets
  – etc.
• i.e., for frequent itemsets of size $\leq k$,
  – $k$ passes over the database are required
FP-growth Algorithm

• An alternative method for finding frequent itemsets
  – usually faster than Apriori
  – requires at most two passes over the database

• Use a compressed representation of the database using an FP-tree

• Once an FP-tree has been constructed, it uses a recursive divide-and-conquer approach to mine the frequent itemsets
FP-Tree Construction

### After reading TID=1:

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{A,B}</td>
</tr>
<tr>
<td>2</td>
<td>{B,C,D}</td>
</tr>
<tr>
<td>3</td>
<td>{A,C,D,E}</td>
</tr>
<tr>
<td>4</td>
<td>{A,D,E}</td>
</tr>
<tr>
<td>5</td>
<td>{A,B,C}</td>
</tr>
<tr>
<td>6</td>
<td>{A,B,C,D}</td>
</tr>
<tr>
<td>7</td>
<td>{B,C}</td>
</tr>
<tr>
<td>8</td>
<td>{A,B,C}</td>
</tr>
<tr>
<td>9</td>
<td>{A,B,D}</td>
</tr>
<tr>
<td>10</td>
<td>{B,C,E}</td>
</tr>
</tbody>
</table>

### After reading TID=2:

The tree is updated as follows:

- After reading TID=1:
  - The root is null, and the path `A:1` is added.
  - The path `B:1` is added.

- After reading TID=2:
  - The path `B:1` is replaced with `B:2`.
  - The path `C:1` is added.
  - The path `D:1` is added.

The tree structure is adjusted accordingly.
FP-Tree Construction

After reading TID=3:

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{A,B}</td>
</tr>
<tr>
<td>2</td>
<td>{B,C,D}</td>
</tr>
<tr>
<td>3</td>
<td>{A,C,D,E}</td>
</tr>
<tr>
<td>4</td>
<td>{A,D,E}</td>
</tr>
<tr>
<td>5</td>
<td>{A,B,C}</td>
</tr>
<tr>
<td>6</td>
<td>{A,B,C,D}</td>
</tr>
<tr>
<td>7</td>
<td>{B,C}</td>
</tr>
<tr>
<td>8</td>
<td>{A,B,C}</td>
</tr>
<tr>
<td>9</td>
<td>{A,B,D}</td>
</tr>
<tr>
<td>10</td>
<td>{B,C,E}</td>
</tr>
</tbody>
</table>

counter is increased
FP-Tree Construction

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{A,B}</td>
</tr>
<tr>
<td>2</td>
<td>{B,C,D}</td>
</tr>
<tr>
<td>3</td>
<td>{A,C,D,E}</td>
</tr>
<tr>
<td>4</td>
<td>{A,D,E}</td>
</tr>
<tr>
<td>5</td>
<td>{A,B,C}</td>
</tr>
<tr>
<td>6</td>
<td>{A,B,C,D}</td>
</tr>
<tr>
<td>7</td>
<td>{B,C}</td>
</tr>
<tr>
<td>8</td>
<td>{A,B,C}</td>
</tr>
<tr>
<td>9</td>
<td>{A,B,D}</td>
</tr>
<tr>
<td>10</td>
<td>{B,C,E}</td>
</tr>
</tbody>
</table>

Transaction Database

Pointers are used to assist frequent itemset generation

Header table

<table>
<thead>
<tr>
<th>Item</th>
<th>Pointer</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
</tr>
</tbody>
</table>
FP-Tree Construction

• Properties of the FP-Tree
  – a very compact representation
  – fits in memory
    • even for larger transaction databases
    • more transactions of the same kind do not increase the tree size
  – can be optimized
    • sorting most frequent items first
    • good compression for many similar transactions
    • pruning all leaf nodes smaller than minSupport
Finding Patterns

• Traverse tree for each item
  – use pointers and sum counts
  – move up the tree
• E.g., minsup=3
  – itemsets ending in E:
    \{E\}, \{CE\}, \{BCE\}

WARNING: simplified illustration!
Finding Patterns

- Traverse tree for each item
  - use pointers and sum counts
  - move up the tree
- E.g., minsup=3
  - itemsets ending in D:
    - \{D\}, \{AD\},
    - \{BD\}, \{CD\}

WARNING: simplified illustration!
Finding Patterns

- Traverse tree for each item
  - use pointers and sum counts
  - move up the tree
- E.g., minsup=3
  - itemsets ending in C:
    - \{C\}, \{AC\}, \{BC\}, \{ABC\}
- Note:
  - \{CD\} is also frequent
  - but we are only looking for itemsets ending in C

WARNING: simplified illustration!
FP-Growth

• Scans the database only twice:
  – first scan counts all 1-itemsets
    • for ordering by most frequent (more compact tree)
    • and for removing itemsets below minup
  – second scan for constructing the FP-tree

• Finding patterns from the tree
  – illustration was simplified
  – actual algorithm recursively decomposes the tree into smaller subtrees
  – details: see books
Frequent Itemset Generation in Rapidminer

- **FP-Growth**
- **Parameters**
  - **find_min_number_of_itemsets**
  - **positive_value**
  - **min_support**
  - **max_items**
  - **must_contain**
Frequent Itemset Generation in Rapidminer

<table>
<thead>
<tr>
<th>No. of Sets: 22</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
</tbody>
</table>
Creating Association Rules in Rapidminer
Exploring Association Rules in Rapidminer

<table>
<thead>
<tr>
<th>No.</th>
<th>Premises</th>
<th>Conclusion</th>
<th>Support</th>
<th>Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>278</td>
<td>marital-status = Never-married</td>
<td>class = &lt;=50K</td>
<td>0.310</td>
<td>0.957</td>
</tr>
<tr>
<td>266</td>
<td>age = range1 [-∞ - 31.500]</td>
<td>class = &lt;=50K</td>
<td>0.330</td>
<td>0.938</td>
</tr>
<tr>
<td>236</td>
<td>sex = Female</td>
<td>class = &lt;=50K</td>
<td>0.308</td>
<td>0.917</td>
</tr>
<tr>
<td>157</td>
<td>workclass = Private</td>
<td>class = &lt;=50K</td>
<td>0.510</td>
<td>0.775</td>
</tr>
<tr>
<td>154</td>
<td>native-country = United-States, workclass = Private</td>
<td>class = &lt;=50K</td>
<td>0.440</td>
<td>0.751</td>
</tr>
<tr>
<td>153</td>
<td>race = White, workclass = Private</td>
<td>class = &lt;=50K</td>
<td>0.418</td>
<td>0.749</td>
</tr>
<tr>
<td>150</td>
<td>native-country = United-States</td>
<td>class = &lt;=50K</td>
<td>0.646</td>
<td>0.736</td>
</tr>
<tr>
<td>149</td>
<td>native-country = United-States, relationship = Husband</td>
<td>class = &lt;=50K</td>
<td>0.376</td>
<td>0.732</td>
</tr>
<tr>
<td>148</td>
<td>race = White</td>
<td>class = &lt;=50K</td>
<td>0.614</td>
<td>0.721</td>
</tr>
<tr>
<td>147</td>
<td>native-country = United-States, race</td>
<td>class = &lt;=50K</td>
<td>0.556</td>
<td>0.715</td>
</tr>
<tr>
<td>146</td>
<td>sex = Male, workclass = Private</td>
<td>class = &lt;=50K</td>
<td>0.302</td>
<td>0.699</td>
</tr>
</tbody>
</table>
Interestingness Measures

- Association rule algorithms tend to produce too many rules
  - many of them are uninteresting or redundant
  - Redundant if \( \{A,B,C\} \to \{D\} \) and \( \{A,B\} \to \{D\} \)
    have same support & confidence

- Interestingness measures can be used to prune or rank the derived rules

- In the original formulation of association rules, support & confidence are the only interest measures used

- Later, various other measures have been proposed
  - See Tan/Steinbach/Kumar, Chapter 6.7
  - We will have a look at one: Lift
**Drawback of Confidence**

<table>
<thead>
<tr>
<th>Coffee</th>
<th>Coffee</th>
<th>Coffee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tea</td>
<td>15</td>
<td>5</td>
</tr>
<tr>
<td>Tea</td>
<td>75</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>10</td>
</tr>
</tbody>
</table>

**Association Rule: Tea → Coffee**

- Confidence = \( \frac{s(\text{Tea} \rightarrow \text{Coffee})}{s(\text{Tea})} = \frac{15}{20} = 0.75 \)
- but \( s(\text{Coffee}) = 0.9 \)
  - Although confidence is high, rule is misleading
  - \( c(\text{Coffee} \rightarrow \overline{\text{Tea}}) = \frac{75}{80} = 0.9375 \)
The *lift* of an association rule $X \rightarrow Y$ is defined as:

$$Lift(X \rightarrow Y) = \frac{c(X \rightarrow Y)}{s(Y)}$$

- Ratio of confidence to support of consequent

**Interpretation:**

- if $lift > 1$, then $X$ and $Y$ are positively correlated
- if $lift < 1$, then $X$ and $Y$ are negatively correlated
- if $lift = 1$, then $X$ and $Y$ are independent.
Example: Lift

\[
\begin{array}{ccc}
\text{Coffee} & \text{Coffee} \\
\hline
\text{Tea} & 15 & 5 & 20 \\
\text{Tea} & 75 & 5 & 80 \\
\text{Tea} & 90 & 10 & 100 \\
\end{array}
\]

Association Rule: \( \text{Tea} \rightarrow \text{Coffee} \)

\[
c(\text{Tea} \rightarrow \text{Coffee}) = 0.75
\]

\[
s(\text{Coffee}) = 0.9
\]

\[
\Rightarrow \text{Lift} = \frac{0.75}{0.9} = 0.8333 \ (< 1, \therefore \text{negatively associated})
\]
There are lots of measures proposed in the literature. Some measures are good for certain applications, but not for others. Details: see literature (e.g., Tan et al.)

<table>
<thead>
<tr>
<th>#</th>
<th>Measure</th>
<th>Formula</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\phi$-coefficient</td>
<td>$\frac{P(A,B) - P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$</td>
</tr>
</tbody>
</table>
| 2  | Goodman-Kruskal’s (lambda)    | $\sum_{a} \max_{b} P(A|B_a) + \sum_{b} \max_{a} P(A|B_b) - \max_{a} P(A|B_a) - \max_{b} P(B|A_b)$ |}
| 3  | Odds ratio ($\alpha$)         | $\frac{P(A,B)P(A,B) - P(A)P(B)}{P(A,B)P(A,B) - P(A)P(B) + P(A,B)P(A,B)} = \frac{a-1}{a+1}$ |
| 4  | Yule’s Q                      | $\sum P(A,B)P(A,B) - P(A)P(B)P(A,B) = \sqrt{a+1}$                       |
| 5  | Yule’s Y                      | $\sum P(A,B)P(A,B)P(A,B) = \sqrt{a+1}$                                 |
| 6  | Kappa ($\kappa$)              | $\sum P(A,B)P(B|A) - P(A)P(B|A)P(B|A) = \sqrt{a+1}$                       |
| 7  | Mutual Information ($M$)       | $\sum P(A,B)P(B|A)P(B|A) = \sqrt{a+1}$                                 |
| 8  | J-Measure ($J$)               | $\max \left( \frac{P(A,B) \log \frac{P(B|A)}{P(B|\bar{A})} + P(A,B) \log \frac{P(B|\bar{A})}{P(B|A)}}, \frac{P(A,B) \log \frac{P(B|A)}{P(B|\bar{A})} + P(A,B) \log \frac{P(B|\bar{A})}{P(B|A)}} \right)$ |
| 9  | Gini index ($G$)              | $\max \left( \frac{P(A)P(B|A)^2 + P(B|A)^2 + P(A)P(B|\bar{A})^2 + P(B|\bar{A})^2} P(A)^2 - P(B)^2 \right)$ |
| 10 | Support ($s$)                 | $P(A,B) = \max \{ P(B|A), P(A|B) \}$                                    |
| 11 | Confidence ($c$)              | $\max \{ P(B|A), P(A|B) \}$                                           |
| 12 | Laplace ($L$)                 | $\max \left( \frac{NP(A,B)+1}{NP(A)+1}, \frac{NP(A,B)+1}{NP(B)+1} \right) \right)$ |
| 13 | Conviction ($V$)              | $\max \left( \frac{P(A,B)P(B|A) \log \frac{P(B|A)}{P(B|\bar{A})} + P(A,B) \log \frac{P(B|\bar{A})}{P(B|A)}}, \frac{P(A,B) \log \frac{P(B|A)}{P(B|\bar{A})} + P(A,B) \log \frac{P(B|\bar{A})}{P(B|A)}} \right)$ |
| 14 | Interest ($I$)                | $\max \left( \frac{P(A,B)P(B|A) \log \frac{P(B|A)}{P(B|\bar{A})} + P(A,B) \log \frac{P(B|\bar{A})}{P(B|A)}}, \frac{P(A,B) \log \frac{P(B|A)}{P(B|\bar{A})} + P(A,B) \log \frac{P(B|\bar{A})}{P(B|A)}} \right)$ |
| 15 | cosine ($f$)                  | $\frac{P(A,B)P(B|A)}{\sqrt{P(A)P(B)}}$                                |
| 16 | Piatesky-Shapiro’s (P.S)      | $P(A,B)P(A,B) - P(A)P(B)$                                              |
| 17 | Certainty factor ($P$)        | $\max \left( \frac{P(B|A)P(B|A) \log \frac{P(B|A)}{P(B|\bar{A})} + P(A,B) \log \frac{P(B|\bar{A})}{P(B|A)}}, \frac{P(A,B) \log \frac{P(B|A)}{P(B|\bar{A})} + P(A,B) \log \frac{P(B|\bar{A})}{P(B|A)}} \right)$ |
| 18 | Added Value ($AV$)            | $\max (P(B|A) - P(B), P(A|B) - P(A))$, $\max (P(B|A) - P(B), P(A|B) - P(A))$ |
| 19 | Collective strength ($S$)     | $\frac{P(A,B)P(B|A) \log \frac{P(B|A)}{P(B|\bar{A})} + P(A,B) \log \frac{P(B|\bar{A})}{P(B|A)}}, \frac{P(A,B) \log \frac{P(B|A)}{P(B|\bar{A})} + P(A,B) \log \frac{P(B|\bar{A})}{P(B|A)}} \right)$ |
| 20 | Jaccard ($\zeta$)             | $\max (P(A,B)P(B|A) \log \frac{P(B|A)}{P(B|\bar{A})} + P(A,B) \log \frac{P(B|\bar{A})}{P(B|A)}), \frac{P(A,B) \log \frac{P(B|A)}{P(B|\bar{A})} + P(A,B) \log \frac{P(B|\bar{A})}{P(B|A)}} \right)$ |
| 21 | Klugman ($K$)                 | $\sqrt{P(A,B) \max \{ P(B|A) - P(B), P(A|B) - P(A) \}}$}
### Handling Continuous and Categorical Attributes

**How to apply association analysis formulation to non-asymmetric binary variables?**

<table>
<thead>
<tr>
<th>Session Id</th>
<th>Country</th>
<th>Session Length (sec)</th>
<th>Number of Web Pages viewed</th>
<th>Gender</th>
<th>Browser Type</th>
<th>Buy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>USA</td>
<td>982</td>
<td>8</td>
<td>Male</td>
<td>IE</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>China</td>
<td>811</td>
<td>10</td>
<td>Female</td>
<td>Netscape</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>USA</td>
<td>2125</td>
<td>45</td>
<td>Female</td>
<td>Mozilla</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>Germany</td>
<td>596</td>
<td>4</td>
<td>Male</td>
<td>IE</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>Australia</td>
<td>123</td>
<td>9</td>
<td>Male</td>
<td>Mozilla</td>
<td>No</td>
</tr>
</tbody>
</table>

**Example of Association Rule:**

\[
\{\text{Number of Pages} \in [5, 10) \land (\text{Browser} = \text{Mozilla})\} \rightarrow \{\text{Buy} = \text{No}\}
\]
Handling Categorical Attributes

- Transform categorical attribute into asymmetric binary variables
- Introduce a new “item” for each distinct attribute-value pair
  - Example: replace Browser Type attribute with
    - Browser Type = Internet Explorer
    - Browser Type = Mozilla
Handling Categorical Attributes

• Potential Issues
  – What if attribute has many possible values
    • Many of the attribute values may have very low support
    • Potential solution: Aggregate the low-support attribute values
  – What if distribution of attribute values is highly skewed
    • Example: 95% of the visitors have Buy = No
    • Most of the items will be associated with (Buy=No) item
    • Potential solution: drop the highly frequent items
Handling Continuous Attributes

- Transform continuous attribute into binary variables using discretization
  - Equal-width binning
  - Equal-frequency binning

- Issue: Size of the intervals affects support & confidence

\[
\{\text{Refund} = \text{No}, \ (\text{Income} = 51,250)\} \rightarrow \{\text{Cheat} = \text{No}\}
\]
\[
\{\text{Refund} = \text{No}, \ (60K \leq \text{Income} \leq 80K)\} \rightarrow \{\text{Cheat} = \text{No}\}
\]
\[
\{\text{Refund} = \text{No}, \ (0K \leq \text{Income} \leq 1B)\} \rightarrow \{\text{Cheat} = \text{No}\}
\]

- Too small intervals: not enough support
- Too large intervals: not enough confidence
Effect of Support Distribution

• Many real data sets have skewed support distribution

![Support distribution of a retail data set](image)
Effect of Support Distribution

• How to set the appropriate *minsup* threshold?
  – If *minsup* is set too high, we could miss itemsets involving interesting rare items (e.g., expensive products)
  – If *minsup* is set too low, it is computationally expensive and the number of itemsets is very large

• Using a single minimum support threshold may not be effective
Multiple Minimum Support

• How to apply multiple minimum supports?
  – MS(i): minimum support for item i
  – e.g.: MS(Milk) = 5%, MS(Coke) = 3%, MS(Broccoli) = 0.1%, MS(Salmon) = 0.5%
  – MS(\{Milk, Broccoli\}) = \min(MS(Milk), MS(Broccoli)) = 0.1%
  – Challenge: Support is no longer anti-monotone
    • Suppose: Support(Milk, Coke) = 1.5% and Support(Milk, Coke, Broccoli) = 0.5%
      \{Milk, Coke\} is infrequent but \{Milk, Coke, Broccoli\} is frequent
    – Requires variations of Apriori algorithm
    – Details: see literature
Wrap-up

• Association Analysis:
  – discovering patterns in data
  – patterns are described by rules

• Apriori algorithm:
  – Finds rules with minimum support (i.e., number of transactions)
  – and minimum confidence (i.e., strength of the implication)

• You'll play around with it in the upcoming exercise...
Questions?