Data Mining I

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

**Goal:** Find co-occurrence relationships, called associations, amongst data items.

- proposed by Agrawal et al. in 1993.
- initially used for **Shopping Basket Analysis** to find how items purchased by customers are related.
- later extended to more complex data structures
  - **sequential patterns**
  - **subgraph patterns**
- and other application domains
  - **web usage mining**
  - **social science**
  - **life science**
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.

**Shopping 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**

\[
\{\text{Diaper}\} \rightarrow \{\text{Beer}\} \\
\{\text{Milk, Bread}\} \rightarrow \{\text{Eggs, Coke}\} \\
\{\text{Beer, Bread}\} \rightarrow \{\text{Milk}\}
\]

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. \(\sigma(\{\text{Milk, Bread, Diaper}\}) = 2\)

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

- Frequent Itemset
  - An itemset whose support is greater than or equal to a \textit{minsup} threshold.

<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>
**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.
  - Example: 
    $\{\text{Milk, Diaper}\} \rightarrow \{\text{Beer}\}$

- **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 $X$

**Examples:**

<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>

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

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

- Given a set of transactions $T$, the goal of association rule mining is to find all rules having
  1. support $\geq \text{minsup}$ threshold
  2. 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

Example of Rules:

{Milk, Diaper} → {Beer} (s=0.4, c=0.67)
{Milk, Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper, Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk, Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk, Beer} (s=0.4, c=0.5)
{Milk} → {Diaper, Beer} (s=0.4, c=0.5)

Observations:

- All the above rules are binary partitions of the same itemset: {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.
Mining Association Rules

- Two-step approach:

1. Frequent Itemset Generation
   - Generate all itemsets whose support $\geq$ minsup

2. Rule Generation
   - Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

- Frequent itemset generation is still computationally expensive.
2. Frequent Itemset Generation

Given $d$ items, there are $2^d$ candidate itemsets!
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

<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>

- Complexity $\sim O(NMw)$ ➔ Expensive since $M = 2^d$ !!!
- A smarter algorithm is required.
Example: 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
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.

**Items** (1-itemsets)

**Pairs** (2-itemsets)

<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} as count {Milk, Beer} = 2.

**Triplets** (3-itemsets)

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Bread, Milk, Diaper}</td>
<td>3</td>
</tr>
</tbody>
</table>
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
   
   2. **Prune** candidate itemsets that can not be frequent because they contain subsets of length $k$ that are infrequent (Apriori Principle)

   3. **Count** the support of each candidate by scanning the DB

   4. **Eliminate** candidates that are infrequent, leaving only those that are frequent
Example: Finding frequent itemsets

<table>
<thead>
<tr>
<th>itemset</th>
<th>count</th>
<th>minsup=0.5</th>
</tr>
</thead>
</table>

1. scan T

- **Cand₁**: \{1\}:2, \{2\}:3, \{3\}:3, \{4\}:1, \{5\}:3
- **Fequ₁**: \{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
- **Fequ₂**: \{1,3\}:2, \{2,3\}:2, \{2,5\}:3, \{3,5\}:2
- **Cand₃**: \{2,3,5\}

3. scan T

- **C₃**: \{2,3,5\}:2
- **F₃**: \{2,3,5\}

**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>
3. Rule Generation

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

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

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

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

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

  \[
  
  \begin{align*}
  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, & & & & 
  \end{align*}
  \]

- 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 with respect to the number of
    items on the right hand side 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:
    \[ c(AB \rightarrow C) := \frac{s(ABC)}{s(AB)} \quad c(A \rightarrow BC) := \frac{s(ABC)}{s(A)} \]
    
  – 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

- BCD => A
- ACD => B
- ABD => C
- ABC => D
- CD => AB
- BD => AC
- BC => AD
- AD => BC
- AC => BD
- AB => CD
- D => ABC
- C => ABD
- B => ACD
- A => BCD

Low Confidence Rule

Pruned Rules
Rule Generation for Apriori Algorithm

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

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

- Prune rule $D \Rightarrow ABC$ if its subset $AD \Rightarrow BC$ does not have high confidence

- All the required information for confidence computation has already been recorded in itemset generation. Thus, there is no need to see the data $T$ any more.
**FP-Growth**

Alternative frequent itemset generation algorithm which compresses data as a tree structure in memory.

Details: Tan/Steinback/Kumar

Chapter 6.6
Creating Association Rules in Rapidminer

- **FP-Growth**
- **Create Association Rules**

Options available:
- **criterion**: confidence
- **min confidence**: 0.1

Note: 2 hidden expert parameters
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, world</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, race</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>
4. Interestingness Measures

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

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

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

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

#### Association Rule: Tea → Coffee

<table>
<thead>
<tr>
<th></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>

Confidence = $P(\text{Coffee}|\text{Tea}) = 0.75$

but $P(\text{Coffee}) = 0.9$

Although confidence is high, rule is misleading as the fraction of coffee drinkers is higher than the confidence of the rule.
Lift

- The *lift* of an association rule $X \rightarrow Y$ is defined as:

$$Lift = \frac{P(Y | X)}{P(Y)}$$

- Ratio of confidence to expected confidence

- 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

Association Rule: Tea → Coffee

Confidence = P(Coffee | Tea) = 0.75

but P(Coffee) = 0.9

⇒ Lift = 0.75/0.9 = 0.8333 (< 1, therefore is negatively correlated)
5. 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 an 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
    - attribute: Browser Type = Internet Explorer
    - attribute: Browser Type = Mozilla
    - ..... 

- 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 discretized intervals affect support & confidence

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

  - If intervals too small
    - may not have enough support
  - If intervals too large
    - may not have enough confidence
Conclusion

- The algorithm does the counting for you and finds patterns in the data.

- You need to do the interpretation based on your knowledge about the application domain.
  - Which patterns are meaningful?
  - Which patterns are surprising?
To find out if two items $x$ and $y$ are bought together, we can also compute their correlation.

**Shortcoming**: Covers only correlation between two items, not between multiple items, e.g. \{Beer, Bread\} $\rightarrow$ \{Milk\}

E.g., using 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](image)

<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 Tabl</th>
<th>Netbook-Sc</th>
<th>HP CE50 T</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>-0.327</td>
<td>0.356</td>
<td>-0.535</td>
<td>1</td>
<td>-0.089</td>
<td>-0.655</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 Tabl</td>
<td>0.583</td>
<td>-0.583</td>
<td>0.356</td>
<td>0.816</td>
<td>0.102</td>
<td>1</td>
<td>-0.667</td>
<td>0.356</td>
<td>0.250</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 T</td>
<td>0.356</td>
<td>-0.356</td>
<td>1</td>
<td>-0.218</td>
<td>-0.327</td>
<td>0.356</td>
<td>-0.535</td>
<td>1</td>
<td>-0.089</td>
<td>-0.655</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>