Sales Transactions

<table>
<thead>
<tr>
<th>TID</th>
<th>Transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Beef, Chicken, Milk</td>
</tr>
<tr>
<td>2</td>
<td>Beef, Cheese</td>
</tr>
<tr>
<td>3</td>
<td>Cheese, Boots</td>
</tr>
<tr>
<td>4</td>
<td>Beef, Chicken, Cheese</td>
</tr>
<tr>
<td>5</td>
<td>Beef, Chicken, Clothes, Cheese, Milk</td>
</tr>
<tr>
<td>6</td>
<td>Chicken, Clothes, Milk</td>
</tr>
<tr>
<td>7</td>
<td>Chicken, Clothes, Milk</td>
</tr>
<tr>
<td>8</td>
<td>Beef, Milk</td>
</tr>
</tbody>
</table>

Support Count

- The support count, or frequency, of an itemset is the number of the transactions that contain the itemset
- Item, Itemset, and Transaction Example:
  - support_count({beef}) = 5
  - support_count({beef, chicken, milk}) = ??

Frequent Itemset

- An itemset is frequent if its support count is greater than or equal to a minimum support count threshold
- support_count(X) ≥ \( \min\_sup \)

The Need for Closed Frequent Itemsets

- Two transactions
  - \(<a_1, a_2, ..., a_i> \) and \(<a_1, a_2, ..., a_i>\)
  - \( \min\_sup = 1 \)
- # of frequent itemsets = ??

Closed Frequent Itemset

- An itemset \( X \) is closed if there exists no proper superset of \( X \) that has the same support count
- A closed frequent itemset is an itemset that is both closed and frequent
Closed Frequent Itemset
Example
ał Two transactions
• \(<a_1, a_2, ..., a_{100}>\) and \(<a_1, a_2, ..., a_{50}>\)
ał min_sup=1
ał Closed frequent itemset(s)??

Maximal Frequent Itemset
Example
ał Two transactions
• \(<a_1, a_2, ..., a_{100}>\) and \(<a_1, a_2, ..., a_{50}>\)
ał min_sup=1
ał Closed frequent itemset(s)??


Maximal Frequent Itemset
Example
ał An itemset \(X\) is a maximal frequent itemset if \(X\) is frequent and there exists no proper superset of \(X\) that is also frequent
ał Example: if \({a, b, c}\) is a maximal frequent itemset, which one of these cannot be a MFI
• \({a, b, c, d}\), \({a, c}\), \({b, d}\)

From Frequent Itemsets to Association Rules
ał \({\text{chicken, milk}}\) is a frequent set
ał \({\text{chicken}}\) \(\Rightarrow\) \({\text{milk}}\)??
ał Or is it \({\text{milk}}\) \(\Rightarrow\) \({\text{chicken}}\)??

Association Rules
ał \(A \Rightarrow B\)
• \(A\) and \(B\) are itemsets
• \(A \cap B = \emptyset\)

Support
ał The support of \(A \Rightarrow B\) is the percentage of the transactions that contain \(A \cup B\)
ał \(\text{support}(A \Rightarrow B) = \frac{\text{support}(A \cup B)}{|D|}\)
ał \(D\) is the set of the transactions

ał \(P(A \cup B)\) is the probability that a transaction contains \(A \cup B\)
Confidence

- The confidence of \( A \Rightarrow B \) is the percentage of the transactions containing \( A \) that also contains \( B \)

\[
\text{confidence}(A \Rightarrow B) = \frac{P(B | A)}{P(A)} = \frac{\text{support}(A \cup B)}{\text{support}(A)}
\]

Support and Confidence Example

- \( \{\text{chicken}\} \Rightarrow \{\text{milk}\} \)?
- \( \{\text{milk}\} \Rightarrow \{\text{chicken}\} \)??

Strong Association Rule

- An association rule is strong if it satisfies both a minimum support threshold (\( \text{min\_sup} \)) and a minimum confidence threshold (\( \text{min\_conf} \))
- Why do we need both \text{support} and \text{confidence}??

Association Rule Mining

- Find strong association rules
  - Find all frequent itemsets
  - Generate strong association rules from the frequent itemsets

The Apriori Property

- All nonempty subsets of a frequent itemset must also be frequent
- Or, if an itemset is not frequent, its supersets cannot be frequent either

Finding Frequent Itemsets – The Apriori Algorithm

- Given \( \text{min\_sup} \)
- Find the frequent 1-itemsets \( L_1 \)
- Find the the frequent k-itemsets \( L_k \) by joining the itemsets in \( L_{k-1} \)
- Stop when \( L_k \) is empty
Apriori Algorithm Example

<table>
<thead>
<tr>
<th>TID</th>
<th>Transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1, 2, 3</td>
</tr>
<tr>
<td>2</td>
<td>1, 4</td>
</tr>
<tr>
<td>3</td>
<td>4, 5</td>
</tr>
<tr>
<td>4</td>
<td>1, 2, 4</td>
</tr>
<tr>
<td>5</td>
<td>1, 2, 6, 4, 3</td>
</tr>
<tr>
<td>6</td>
<td>2, 6, 3</td>
</tr>
<tr>
<td>7</td>
<td>2, 6, 3</td>
</tr>
<tr>
<td>8</td>
<td>1, 3</td>
</tr>
</tbody>
</table>

◆ Support 25%

L1

- Scan the data once to get the count of each item
- Remove the items that do not meet min_sup

<table>
<thead>
<tr>
<th>C</th>
<th>support_count</th>
<th>L1</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>5</td>
<td>(1)</td>
</tr>
<tr>
<td>(2)</td>
<td>5</td>
<td>(2)</td>
</tr>
<tr>
<td>(3)</td>
<td>5</td>
<td>(3)</td>
</tr>
<tr>
<td>(4)</td>
<td>4</td>
<td>(4)</td>
</tr>
<tr>
<td>(5)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(6)</td>
<td>3</td>
<td>(6)</td>
</tr>
</tbody>
</table>

L2

- \( C_2 = L_1 \times L_1 \)
- Scan the dataset again for the support_count of \( C_2 \), then remove non-frequent itemsets from \( C_2 \), i.e. \( C_2 \rightarrow L_2 \)

<table>
<thead>
<tr>
<th>C</th>
<th>support_count</th>
<th>L2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1,2)</td>
<td>3</td>
<td>(1,2)</td>
</tr>
<tr>
<td>(1,3)</td>
<td>3</td>
<td>(1,3)</td>
</tr>
<tr>
<td>(1,4)</td>
<td>3</td>
<td>(1,4)</td>
</tr>
<tr>
<td>(1,6)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(2,3)</td>
<td>4</td>
<td>(2,3)</td>
</tr>
<tr>
<td>(2,4)</td>
<td>2</td>
<td>(2,4)</td>
</tr>
<tr>
<td>(2,6)</td>
<td>3</td>
<td>(2,6)</td>
</tr>
<tr>
<td>(3,4)</td>
<td>1</td>
<td>(3,4)</td>
</tr>
<tr>
<td>(3,6)</td>
<td>3</td>
<td>(3,6)</td>
</tr>
<tr>
<td>(4,6)</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

L3

◆ ??

From \( L_{k-1} \) to \( C_k \)

- Let \( l_1 \) be an itemset in \( L_{k-1} \), and \( l_1[j] \) be the \( j \)th item in \( l_1 \)
- Items in an itemset are sorted, i.e. \( l_1[1] < l_1[2] < ... < l_1[k-1] \)
- \( l_1 \) and \( l_2 \) are joinable if
  - Their first \( k-2 \) items are the same, and
  - \( l_1[k-1] < l_2[k-2] \)

From \( C_k \) to \( L_k \)

- Reduce the size of \( C_k \) using the Apriori property
  - any \((k-1)\)-subset of an candidate must be frequent, i.e. in \( L_{k-1} \)
- Scan the dataset to get the support counts
Generate Association Rules from Frequent Itemsets

- For each frequent itemset \( l \), generate all nonempty subset of \( l \).
- For every nonempty subset of \( s \) of \( l \), output rule \( s \Rightarrow (l-s) \) if \( \text{conf}(s \Rightarrow (l-s)) \geq \min \text{conf} \).

Confidence-based Pruning ...

- \( \text{conf}(\{a,b\} \Rightarrow \{c,d\}) < \min \text{conf} \)
  - \( \text{conf}(\{a\} \Rightarrow \{c,d\}) ?? \)
  - \( \text{conf}(\{a,b,e\} \Rightarrow \{c,d\}) ?? \)

... Confidence-based Pruning

- If \( \text{conf}(s \Rightarrow (l-s)) < \min \text{conf} \), then
  - \( \text{conf}(s' \Rightarrow (l-s')) < \min \text{conf} \)
  where \( s' \subseteq s \).
- Example:
  - \( \text{conf}(\{a,b\} \Rightarrow \{c,d\}) < \min \text{conf} \)
  - ??

Limitations of the Apriori Algorithm

- Multiple scans of the datasets
  - How many??
- Need to generate a large number of candidate sets

Partitioning

- Divide dataset into \( n \) non-overlapping partitions such that each partition fits into main memory.
- Find local frequent itemsets in each partition with \( \min \text{sup} \) (1 scan).
- All local frequent itemsets form a candidate set.
  - Does it include all global frequent itemsets??
- Find global frequent itemsets from candidates (1 scan).

FP-Growth Algorithm

- Frequent-pattern Growth
- Mine frequent itemsets without candidate generation.
FP-Growth Example

<table>
<thead>
<tr>
<th>TID</th>
<th>Transactions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{I1, I2, I5}</td>
</tr>
<tr>
<td>2</td>
<td>{I2, I4}</td>
</tr>
<tr>
<td>3</td>
<td>{I2, I3}</td>
</tr>
<tr>
<td>4</td>
<td>{I1, I2, I4}</td>
</tr>
<tr>
<td>5</td>
<td>{I1, I3}</td>
</tr>
<tr>
<td>6</td>
<td>{I2, I3}</td>
</tr>
<tr>
<td>7</td>
<td>{I1, I3}</td>
</tr>
<tr>
<td>8</td>
<td>{I1, I2, I3, I5}</td>
</tr>
<tr>
<td>9</td>
<td>{I1, I2, I3}</td>
</tr>
</tbody>
</table>

min_sup = 2

L

- Scan the dataset and find the frequent 1-itemsets
- Sort the 1-itemsets by support count in descending order

I2: 7
I1: 6
I3: 6
I4: 2
I5: 2

FP Tree

- Each transaction is processed in L order (why??) and becomes a branch in the FP tree
- Each node is linked from L

FP Tree Construction ...

T1: {I2, I1, I5}

T2: {I2, I4}

... FP Tree Construction ...

... FP Tree Construction
Mining the FP Tree

- For each item $i$ in $L$ (in ascending order), find the branch(s) in the FP tree that ends in $i$.
- If there’s only one branch, generate the frequent itemsets that end in $i$; otherwise run the tree mining algorithm recursively on the subtree.

Mining the FP Tree – $I_5$

Mining The FP Tree – $I_3$ ...

... Mining The FP Tree – $I_3$

Mining Frequent Itemsets Using Vertical Data Format

<table>
<thead>
<tr>
<th>Itemset</th>
<th>TID_set</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>T1, T4, T5, T7, T8, T9</td>
</tr>
<tr>
<td>12</td>
<td>T1, T2, T3, T4, T6, T8, T9</td>
</tr>
<tr>
<td>13</td>
<td>T3, T5, T6, T7, T8, T9</td>
</tr>
<tr>
<td>14</td>
<td>T2, T4</td>
</tr>
<tr>
<td>15</td>
<td>T1, T8</td>
</tr>
</tbody>
</table>

And then what??

Strong Association Rules Could Be Misleading ...

Example:
- 10,000 transactions
- 6,000 transactions included games
- 7,500 transactions included videos
- 4,000 transactions included both

{game} $\Rightarrow$ {video}
- Support?? Confidence??
... Strong Association Rules Could Be Misleading

Does buying game really imply buying video as well??

Correlation Measures for Association Rules

- Lift
- $\chi^2$
- All_confidence
- Cosine

Lift

$$\text{lift}(A, B) = \frac{P(A \cup B)}{P(A)P(B)}$$

- $A$ and $B$ are
  - Independent if $\text{lift}(A, B) = 1$
  - Correlated if $\text{lift}(A, B) > 1$
  - Negatively correlated if $\text{lift}(A, B) < 1$
- $\text{lift}([\text{game}],[\text{video}]) = ??$

$\chi^2$

Two attributes $A$ and $B$
- $A$ has $r$ possible values
- $B$ has $c$ possible values
- Event: $(A=a_i, B=b_j)$
- Observed frequency: $o_{ij}$
- Expected frequency:
  - $e_{ij} = \frac{\text{count}(A=a_i) \times \text{count}(B=b_j)}{N}$

$$\chi^2 = \sum_{i=1}^{r} \sum_{j=1}^{c} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

$\chi^2$ Example – Observed Frequency

<table>
<thead>
<tr>
<th></th>
<th>male</th>
<th>female</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>fiction</td>
<td>250</td>
<td>200</td>
<td>450</td>
</tr>
<tr>
<td>non-fiction</td>
<td>50</td>
<td>1000</td>
<td>1050</td>
</tr>
<tr>
<td>total</td>
<td>300</td>
<td>1200</td>
<td>1500</td>
</tr>
</tbody>
</table>
**χ² Example – Expected Frequency**

<table>
<thead>
<tr>
<th></th>
<th>male</th>
<th>female</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>fiction</td>
<td>??</td>
<td>??</td>
<td>450</td>
</tr>
<tr>
<td>non-fiction</td>
<td>??</td>
<td>??</td>
<td>1050</td>
</tr>
<tr>
<td>total</td>
<td>300</td>
<td>1200</td>
<td>1500</td>
</tr>
</tbody>
</table>

**Contingency Table and χ²**

<table>
<thead>
<tr>
<th></th>
<th>male</th>
<th>female</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>fiction</td>
<td>250(90)</td>
<td>200(360)</td>
<td>450</td>
</tr>
<tr>
<td>non-fiction</td>
<td>50(210)</td>
<td>1000(840)</td>
<td>1050</td>
</tr>
<tr>
<td>total</td>
<td>300</td>
<td>1200</td>
<td>1500</td>
</tr>
</tbody>
</table>

\[ χ^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93 \]

**χ² Test**

- Degree of freedom \( k = (r-1) \times (c-1) \)
- Significance probability level < 0.05

**Exercise**

- The game and video example
  - Two attributes: buy game, buy video
  - Values of "buy game"? Values of "buy video"?
  - Contingency table?
  - Degree of freedom?
  - \( χ^2 \)??

**All_confidence**

\[ X = \{i_1, i_2, ..., i_k\} \]

\[ all_{-}conf(X) = \sup(X) \frac{\sup(X)}{\max_{\text{item}} \sup(X)} = \sup(X) \frac{\sup(X)}{\max \{\sup(i) \mid i \in X\}} \]

**All_confidence Example**

- Two attributes \( A \) and \( B \)
- \( all_{-}conf(A, B) \)
  - If \( A \) and \( B \) are completely positively correlated
  - If \( A \) and \( B \) are completely negatively correlated
  - If \( A \) and \( B \) are independent
Cosine Measure

\[
\cosine(A, B) = \frac{P(A \cup B)}{\sqrt{P(A) \times P(B)}} = \frac{\sup(A \cup B)}{\sqrt{\sup(A) \times \sup(B)}}
\]

Cosine vs. Lift

\[
lift(A, B) = \frac{P(A \cup B)}{\frac{N}{P(A) \times \sup(B)}} = \frac{N \sup(A \cup B)}{\sup(A) \times \sup(B)}
\]

Choosing Correlation Measures ...

<table>
<thead>
<tr>
<th>datasets</th>
<th>mc</th>
<th>m’c</th>
<th>m’c’</th>
<th>all_confidence</th>
<th>cosine</th>
<th>lift</th>
<th>(\chi^2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1,000</td>
<td>100</td>
<td>100</td>
<td>100,000</td>
<td>0.91</td>
<td>0.91</td>
<td>83.64</td>
</tr>
<tr>
<td>A</td>
<td>1,000</td>
<td>100</td>
<td>100</td>
<td>10,000</td>
<td>0.91</td>
<td>0.91</td>
<td>9.26</td>
</tr>
<tr>
<td>A</td>
<td>1,000</td>
<td>100</td>
<td>100</td>
<td>100</td>
<td>0.91</td>
<td>0.91</td>
<td>1.82</td>
</tr>
<tr>
<td>A</td>
<td>1,000</td>
<td>100</td>
<td>100</td>
<td>0</td>
<td>0.91</td>
<td>0.91</td>
<td>0.99</td>
</tr>
<tr>
<td>B</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>1,000</td>
<td>0.50</td>
<td>0.50</td>
<td>1.00</td>
</tr>
</tbody>
</table>

mc: # of transactions that contain both milk and coffee
m’c: # of transactions that contain neither milk nor coffee

... Choosing Correlation Measures

- all_confidence and cosine are null-invariant, while lift and \(\chi^2\) are not
- all_confidence has the Apriori property
- all_confidence and cosine should be augmented with other measures when the result is not conclusive

Mining Sequential Patterns

- \(<\{\text{computer}\},\{\text{printer}\},\{\text{printer cartridge}\}>\)
- \(<\{\text{bread,milk}\},\{\text{bread,milk}\},\{\text{bread,milk}\}>\)
- \(<\{\text{home.jsp}\},\{\text{search.jsp}\},\{\text{product.jsp}\}>\)
- \(<\{\text{product.jsp}\},\{\text{search.jsp}\}>\)

Terminology and Notations

- Item, itemset
- Event = itemset
- A sequence is an ordered list of events
  - \(<e_1,e_2,e_3>\)
  - E.g. \(<(a)(abc)(bc)(d)(ac)(f)>\)
- The length of a sequence is the number of items in the sequence, i.e. not the number of events
Sequences vs. Itemsets
- \{a,b,c\}
  - \# of 3-itemset(s)?
  - \# of 3-sequence(s)?

Subsequence
- \( A = <a_1, a_2, a_3, ..., a_n> \)
- \( B = <b_1, b_2, b_3, ..., b_m> \)
- \( A \) is a *subsequence* of \( B \) if there exists \( 1 \leq j_1 < j_2 < ... < j_n \leq m \) such that \( a_1 \subseteq b_{j_1}, a_2 \subseteq b_{j_2}, ..., a_n \subseteq b_{j_n} \)

Subsequence Example
- \( s = <(abc)(de)(f)> \)
- What are the subsequences of \( s \)?

Sequential Pattern
- If \( A \) is a subsequence of \( B \), we say \( B \) contains \( A \)
- The support count of \( A \) is the number of sequences that contain \( A \)
- \( A \) is frequent if \( \text{support count}(A) \geq \text{min_sup} \)
- A frequent sequence is called a sequential pattern

Apriori Property Again
- Every nonempty subsequence of a frequent sequence is frequent

GSP Algorithm
- Generalized Sequential Patterns
- An extension of the Apriori algorithm for mining sequential patterns
**GSP Example**

<table>
<thead>
<tr>
<th>SID</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;(a)(ab)(a)&gt;</td>
</tr>
<tr>
<td>2</td>
<td>&lt;(a)(c)(bc)&gt;</td>
</tr>
<tr>
<td>3</td>
<td>&lt;(ab)(c)(b)&gt;</td>
</tr>
<tr>
<td>4</td>
<td>&lt;(a)(c)(c)&gt;</td>
</tr>
</tbody>
</table>

min_sup = 2

**L₁**

<table>
<thead>
<tr>
<th>C₁</th>
<th>support_count</th>
<th>L₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>4</td>
<td>&lt;(a)&gt;</td>
</tr>
<tr>
<td>b</td>
<td>3</td>
<td>&lt;(b)&gt;</td>
</tr>
<tr>
<td>c</td>
<td>3</td>
<td>&lt;(c)&gt;</td>
</tr>
</tbody>
</table>

**L₂**

<table>
<thead>
<tr>
<th>C₂</th>
<th>support_count</th>
<th>L₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;(a)&gt;</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>&lt;(b)&gt;</td>
<td>3</td>
<td>&lt;(ab)&gt;</td>
</tr>
<tr>
<td>&lt;(a)c&gt;</td>
<td>3</td>
<td>&lt;(ac)&gt;</td>
</tr>
<tr>
<td>&lt;(b)&gt;</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>&lt;(b)c&gt;</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>&lt;(c)&gt;</td>
<td>0</td>
<td>&lt;(c)&gt;</td>
</tr>
<tr>
<td>&lt;(c)b&gt;</td>
<td>2</td>
<td>&lt;(cb)&gt;</td>
</tr>
<tr>
<td>&lt;(c)c&gt;</td>
<td>2</td>
<td>&lt;(cc)&gt;</td>
</tr>
<tr>
<td>&lt;(ab)&gt;</td>
<td>2</td>
<td>&lt;(ab)&gt;</td>
</tr>
<tr>
<td>&lt;(ac)&gt;</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>&lt;(bc)&gt;</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

**From L_k⁻¹ to C_k**

- Two sequences s₁ and s₂ are joinable if the subsequence obtained by dropping the first item in s₁ is the same as the subsequence obtained by dropping the last item in s₂.
- The joined sequence is s₁ concatenated with the last item i of s₂:
  - If the last two items in s₂ are in the same event, i is merged into the last event of s₁;
  - Otherwise i becomes a separate event.

**L₃**

<table>
<thead>
<tr>
<th>C₃</th>
<th>support_count</th>
<th>L₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;(a)(c)(b)&gt;</td>
<td>2</td>
<td>&lt;(a)(c)(b)&gt;</td>
</tr>
</tbody>
</table>

**Candidate Pruning**

- A k-sequence can be pruned if one of its (k-1)-subsequence is not frequent.
Summary

- Frequent itemsets, association rules, sequential patterns
  - Measures: support, confidence, correlation
  - Algorithms: Apriori, FP-Growth, vertical data format, rule generation, GPS