Support Count

- The support count, or frequency, of a itemset is the number of the transactions that contain the itemset
  - Item, Itemset, and Transaction
- Examples:
  - \( \text{support\_count}\{\text{beef}\} = 5 \)
  - \( \text{support\_count}\{\text{beef, chicken, milk}\} = ?? \)

Frequent Itemset

- An itemset is frequent if its support count is greater than or equals to a minimum support count threshold
  - \( \text{support\_count}\{X\} \geq \text{min\_sup} \)

The Need for Closed Frequent Itemsets

- Two transactions
  - \( \langle a_1, a_2, ..., a_{100} \rangle \) and \( \langle a_1, a_2, ..., a_{80} \rangle \)
  - \( \text{min\_sup} = 1 \)
- \# 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

- Two transactions
  - \( <a_1, a_2, ..., a_{100}> \) and \( <a_1, a_2, ..., a_{50}> \)
- \( \text{min}_\text{sup}=1 \)
- Closed frequent itemset(s)??

Maximal Frequent Itemset

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

Maximal Frequent Itemset Example

- Two transactions
  - \( <a_1, a_2, ..., a_{100}> \) and \( <a_1, a_2, ..., a_{50}> \)
- \( \text{min}_\text{sup}=1 \)
- Maximal frequent itemset(s)??
- Maximal frequent itemset vs. closed frequent itemset??

From Frequent Itemsets to Association Rules

- \( \{\text{chicken}, \text{cheese}\} \) is a frequent set
- \( \{\text{chicken}\} \Rightarrow \{\text{cheese}\}?? \)
- Or is it \( \{\text{cheese}\} \Rightarrow \{\text{chicken}\}?? \)

Association Rules

- \( A \Rightarrow B \)
  - \( A \) and \( B \) are itemsets
  - \( A \cap B = \emptyset \)

Support

- The support of \( A \Rightarrow B \) is the percentage of the transactions that contain \( A \cup B \)

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

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

\( D \) is the set of the transactions
Confidence

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

\[
\text{confidence}(A \Rightarrow B) = \frac{\text{support_count}(A \cup B)}{\text{support_count}(A)}
\]

Support and Confidence Example

- \{chicken\} \( \Rightarrow \) \{cheese\}??
- \{cheese\} \( \Rightarrow \) \{chicken\}??

Strong Association Rule

- An association rule is strong if it satisfies both a minimum support threshold (min_sup) and a minimum confidence threshold (min_conf)
- Why do we need both support and 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 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>Item</th>
<th>TID</th>
</tr>
</thead>
<tbody>
<tr>
<td>T-shirt</td>
<td>1</td>
</tr>
<tr>
<td>Chicken</td>
<td>2</td>
</tr>
<tr>
<td>Milk</td>
<td>3</td>
</tr>
<tr>
<td>Cheese</td>
<td>4</td>
</tr>
<tr>
<td>Boots</td>
<td>5</td>
</tr>
<tr>
<td>Wallet</td>
<td>6</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>Cᵢ</th>
<th>support_count</th>
<th>Lᵢ</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
- C₂ = L₁ × L₁
- Scan the dataset again for the support_count of C₂, then remove non-frequent itemsets from C₂, i.e. C₂ → L₂

L₃
- ??

From Lₖ₋₁ to Cₖ
- Let $l₁$ be an itemset in $L_{k-1}$, and $l₁[j]$ be the jth item in $l₁$
- Items in an itemset are sorted, i.e. $l₁[1]<l₁[2]<...<l₁[k-1]$
- $l₁$ and $l₂$ are joinable if
  - Their first k-2 items are the same, and
  - $l₁[k-1]<l₂[k-1]$

From Cₖ to Lₖ
- Reduce the size of Cₖ 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 \( \text{conf}(s \Rightarrow (l-s)) \geq \text{min}_\text{conf} \)

Confidence-based Pruning ...

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

... Confidence-based Pruning

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

Limitations of the Apriori Algorithm

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

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>11, 12, 15</td>
</tr>
<tr>
<td>2</td>
<td>12, 14</td>
</tr>
<tr>
<td>3</td>
<td>12, 13, 16</td>
</tr>
<tr>
<td>4</td>
<td>11, 12, 14</td>
</tr>
<tr>
<td>5</td>
<td>11, 13</td>
</tr>
<tr>
<td>6</td>
<td>12, 13</td>
</tr>
<tr>
<td>7</td>
<td>11, 13</td>
</tr>
<tr>
<td>8</td>
<td>11, 12, 13, 15</td>
</tr>
<tr>
<td>9</td>
<td>11, 12, 13</td>
</tr>
</tbody>
</table>

\( \text{min}_\text{sup} = 2 \)
L

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

Frequent 1-itemsets:
- I1: 6
- I2: 7
- 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\}

FP-tree Construction ...

T2: \{I2, I4\}

Mining the FP-tree

- For each item \(i\) in \(L\) (in ascending order), find the branch(s) in the FP tree that ends in \(i\) — Prefix Paths
  - If there is only one path, generate all the frequent patterns ended in \(i\)
  - Else create the Conditional FP-tree for \(i\) and recursively run the mining algorithm on the conditional FP-tree
Prefix Paths of I5

I5 is frequent $\Rightarrow \{I5:2\}$

From Prefix Paths to
Conditional FP-tree

$\Rightarrow \text{Adjust the support counts}$

From Prefix Paths to
Conditional FP-tree

$\Rightarrow \text{Remove the suffix}$

From Prefix Paths to
Conditional FP-tree

$\Rightarrow \text{Remove the infrequent items}$

Conditional FP-tree for I5

$\Rightarrow \text{A FP-tree with suffix pattern } \{I5\}$

Prefix Paths of $\{I1,I5\}$

$\Rightarrow \{I1,I5\}$ is frequent and there is a single path
$\Rightarrow \{I1,I5:2\}, \{I2,I1,I5:2\}$
Prefix Paths of \{I2,I5\}

\(\{I2,I5\}\) is frequent and there is a single path

\(\{I2,I5;2\}\)

All Frequent Itemsets with Suffix I5

\(\{I5;2\}\)
\(\{I1,I5;2\}, \{I2,I1,I5;2\}\)
\(\{I2,I5;2\}\)

Mining The FP-tree – I3 ...

Prefix Paths of I3

\(\{I3;6\}\)

Conditional FP-tree with suffix pattern \{I3\}

Prefix Paths of \{I1,I3\}

\(\{I1,I3;4\}\)

Conditional FP-tree for \{I2,I1,I3\}
Mining The FP-tree – I3

All Frequent Itemsets with Suffix I3
- {I3:6}
- {I1,I3:4}
- {I2,I1,I3:2}
- {I2,I3:4}

About FP-tree Mining
- A divide-and-conquer approach
- Patterns with suffix: I5, I4, I3, I1, I2

Data Partitioning
- Divide dataset into non-overlapping partitions such that each partition fits into main memory
- Find local frequent itemsets in each partition (1 scan)
- Local min_supp??
- All local frequent itemsets form a candidate set
  - Will it include all the global frequent itemsets??
- Find global frequent itemsets from candidates (1 scan)

Vertical Data Format
<table>
<thead>
<tr>
<th>Item</th>
<th>TID_set</th>
</tr>
</thead>
<tbody>
<tr>
<td>I1</td>
<td>T1,T4,T5,T7,T8,T9</td>
</tr>
<tr>
<td>I2</td>
<td>T1,T2,T3,T4,T6,T8,T9</td>
</tr>
<tr>
<td>I3</td>
<td>T2,T5,T6,T7,T8,T9</td>
</tr>
<tr>
<td>I4</td>
<td>T2,T4</td>
</tr>
<tr>
<td>I5</td>
<td>T1,T8</td>
</tr>
</tbody>
</table>

And how does it help??
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

\[ \text{lift}(A, B) = \frac{P(AB)}{P(A)P(B)} \]

From Multiplication Rule to Lift

Correlation

\begin{align*}
\text{lift}(\{\text{game}\}, \{\text{video}\}) &= ?? \\
\end{align*}

Multiplication Rule

\text{If two events A and B are independent of each other}

\[ P(AB) = P(A)P(B) \]

Problem of Lift

\begin{align*}
\text{datasets} & | \text{mc} | \text{m’c} | \text{mc’} | \text{m’c’} | \text{total} | \text{lift} \\
A_1 & | 100 | 100 | 100 | 100 | 400 | ?? \\
A_2 & | 100 | 100 | 100 | 1,000 | 1,300 | ?? \\
A_3 & | 100 | 100 | 100 | 10,000 | 10,300 | ?? \\
A_4 & | 100 | 100 | 100 | 100,000 | 100,300 | ?? \\
\end{align*}

\text{mc}: \# \text{ of transactions that contain both milk and coffee}
\text{m’c}: \# \text{ of transactions that contain milk but not coffee}
\text{mc’}: \# \text{ of transactions that contain coffee but not milk}
\text{m’c’}: \# \text{ of transactions that contain neither milk nor coffee
Null-invariant

A null-transaction is a transaction that does not contain any of the itemsets being examined.

A correlation measure is null-invariant if its value is not affected by the number of null-transactions.

Some Null-invariant Measures

- **Kulczynski Measure**
  \[ \text{Kulczynski Measure} = \frac{2}{\text{sup}(A \cup B) + \text{sup}(A \cap B)} \]
- **Max_confidence**
  \[ \text{Max_confidence} = \max \{ P(A | B), P(B | A) \} \]
- **All_confidence**
  \[ \text{All_confidence} = \min \{ P(A | B), P(B | A) \} \]
- **Cosine Measure**
  \[ \text{Cosine Measure} = \frac{P(AB)}{\sqrt{P(A) \times P(B)}} \]

Ranges of Correlation Measures

<table>
<thead>
<tr>
<th>Measure</th>
<th>Perfectly positively correlated</th>
<th>Perfectly negatively correlated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cosine</td>
<td></td>
<td></td>
</tr>
<tr>
<td>All_conf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max_conf</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kulc</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Choosing Correlation Measures...

<table>
<thead>
<tr>
<th>datasets</th>
<th>mc</th>
<th>m'c</th>
<th>mc'</th>
<th>m'c'</th>
<th>cosine</th>
<th>all_conf</th>
<th>max_conf</th>
<th>Kulc</th>
</tr>
</thead>
<tbody>
<tr>
<td>D_1</td>
<td>10,000</td>
<td>1000</td>
<td>1000</td>
<td>100,000</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td></td>
</tr>
<tr>
<td>D_2</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>100,000</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td></td>
</tr>
<tr>
<td>D_3</td>
<td>1,000</td>
<td>1000</td>
<td>1000</td>
<td>100,000</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td></td>
</tr>
<tr>
<td>D_4</td>
<td>1,000</td>
<td>1000</td>
<td>1000</td>
<td>100,000</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td></td>
</tr>
<tr>
<td>D_5</td>
<td>1,000</td>
<td>1000</td>
<td>1000</td>
<td>100,000</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td></td>
</tr>
<tr>
<td>D_6</td>
<td>1,000</td>
<td>10,000</td>
<td>100,000</td>
<td>100,000</td>
<td>0.91</td>
<td>0.91</td>
<td>0.91</td>
<td></td>
</tr>
</tbody>
</table>

... Choosing Correlation Measures...

- **Recommended:**
  - Kulczynski + Imbalance Ratio (IR)
  \[ IR(A, B) = \frac{\left| \text{sup}(A) - \text{sup}(B) \right|}{\text{sup}(A) + \text{sup}(B) - \text{sup}(A \cup B)} \]
Mining Sequential Patterns

- `<computer>, {printer}, {printer cartridge}>`
- `<bread, milk}, {bread, milk}, {bread, milk}>`
- `<home.jsp}, {search.jsp}, {product.jsp}, {product.jsp}, {search.jsp}>`

Terminology and Notations

- **Item**, itemset
- **Event** = itemset
- A sequence is an ordered list of events
  - `<e1, e2, ..., en>`
  - 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** = `<a1, a2, ..., an>`
- **B** = `<b1, b2, ..., bm>`
- **A** is a *subsequence* of **B** if there exists
  - `1 ≤ j1 < j2 < ... < jn ≤ m` such that `a1 ⊆ b1, a2 ⊆ b2, ..., an ⊆ bn`

Subsequence Example

- `s = <(abc)(de)(f)>`
- Which of these are subsequences of `s`?
  - `s1 = <(ab) (d)>`
  - `s2 = <(ab) (f)>`
  - `s3 = <(ac) (f)>`
  - `s4 = <(abcde)>
  - `s5 = <(a) (de)>`
  - `s6 = <(de) (a) (f)>`

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
  - `support_count(A) ≥ 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>
<th>support_count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>&lt;(a)(ab)(a)&gt;</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>&lt;(a)(ab)(b)&gt;</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>&lt;(a)(ab)(bc)&gt;</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>&lt;(a)(c)(c)&gt;</td>
<td>3</td>
</tr>
</tbody>
</table>

GSP Example

1. \(C_1\) = <(a)> with support 4
2. \(C_1\) = <(b)> with support 3
3. \(C_1\) = <(c)> with support 3

\(L_1\) = \{<a>, <b>, <c>\}

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

- Two sequences \(s_1\) and \(s_2\) are joinable if the subsequence obtained by dropping the first item in \(s_1\) is the same as the subsequence obtained by dropping the last item in \(s_2\).
- The joined sequence is \(s_1\) concatenated with the last item \(i\) of \(s_2\):
  - If the last two items in \(s_2\) are in the same event, \(i\) becomes the last event of \(s_1\);
  - Otherwise \(i\) becomes a separate event.
Candidate Pruning

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

Candidate generation → Pruning

L2: Candidate sequence C1: Pruning sequence

- <1)(2)(3)>
- <1)(2)(3)(4)>
- <1)(2 5)(3)>
- <1)(2 5 3)(4)>
- <1)(5)(3 4)>
- "2 5)(3)>
- <2 5)(3 4)>
- <3)(4)(5)>
- <3)(4)(5)>
- <5)(3 4)>

Subgraph Patterns

Applications in web mining, computational chemistry, bioinformatics, network computing ...

Vertices and Edges

Why Labels?

Vertex labels: \{a, b, c\}
Edge labels: \{p, q\}
Subgraph

A graph $G' = (V', E')$ is a subgraph of another graph $G = (V, E)$ if its vertex set $V'$ is a subset of $V$ and its edge set $E'$ is a subset of $E$.

Support

<table>
<thead>
<tr>
<th>Dataset</th>
<th>O</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-C-C-N</td>
<td>C-C-N-C</td>
</tr>
<tr>
<td>C-S-C-C</td>
<td>N-O</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Subgraph Patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>S-C-C-O</td>
</tr>
<tr>
<td>C-C-N</td>
</tr>
<tr>
<td>N</td>
</tr>
</tbody>
</table>

| Support count = ?? | Support count = ?? |

Frequent Subgraph Mining

Given a set of graphs and $\text{min}_\text{sup}$, find all subgraphs $g$ with $\text{support}(g) \geq \text{min}_\text{sup}$

- Typically we only consider graphs that undirected and connected

Transform Graph to Itemset

Each combination of an edge label with its corresponding vertex labels is mapped to an “item”

- $S \rhd C \rhd C \rhd N$
- $S \rhd C \rhd C \rhd N$
- $(1, S, C), (1, C, C), (1, C, N), (2, C, O)$

Problem??

Apriori-based Approach

- Candidate generation
- Candidate pruning
- Support counting
- Candidate elimination

Candidate Generation

Merge two frequent $(k-1)$-subgraphs to form a candidate $k$-subgraph
- What is $k$??
- The two $(k-1)$-subgraphs must share a common $(k-2)$-subgraph, referred to as their core
Vertex Growing

\[ \begin{align*}
A &\quad D \\
B &\quad A
\end{align*} \quad \begin{align*}
+ \\
B &\quad A
\end{align*} \quad \begin{align*}
&\quad C \\
&\quad = \quad ??
\end{align*} \]

Edge Growing ...

\[ \begin{align*}
A &\quad D \\
B &\quad A
\end{align*} \quad \begin{align*}
+ \\
B &\quad A
\end{align*} \quad \begin{align*}
&\quad C \\
&\quad = \quad ??
\end{align*} \]

... Edge Growing

\[ \begin{align*}
A &\quad D \\
B &\quad A
\end{align*} \quad \begin{align*}
+ \\
B &\quad A
\end{align*} \quad \begin{align*}
&\quad D \\
&\quad = \quad ??
\end{align*} \]

Candidate Pruning

\[ \begin{align*}
\text{\textbullet Remove an edge from a candidate k-subgraph and check if the resulting (k-1)-subgraph is connected and frequent}
\end{align*} \]

Graph Isomorphism Problem

\[ \begin{align*}
\text{\textbullet Determine whether two graphs are topologically equivalent, i.e. } \text{isomorphic}
\end{align*} \]

Adjacency Matrix ...

\[ \begin{align*}
\text{A} &\quad 3 \\
1 &\quad B
\end{align*} \quad \begin{align*}
\text{1} &\quad 1 \\
\text{1} &\quad \text{2}
\end{align*} \quad \begin{align*}
\text{3} &\quad 0 \\
\text{0} &\quad \text{0}
\end{align*} \quad \begin{align*}
\Rightarrow \quad \begin{bmatrix}
0 & 1 & 1 & 3 \\
1 & 0 & 2 & 0 \\
1 & 2 & 0 & 0 \\
3 & 0 & 0 & 0
\end{bmatrix}
\end{align*} \]
... Adjacency Matrix

How many adjacency matrices can a graph with k vertices have??

String Representation of an Adjacency Matrix

Graph Code

◆A.K.A. Canonical label
◆The string representation of the adjacency matrix that has the lowest (or highest) lexicographic value

Support Counting

◆Isomorphism test a candidate k-subgraph against the k-subgraphs of each graph

Summary

◆Frequent itemsets, association rules, sequential patterns, subgraph patterns
  ■ Measures: support, confidence, correlation
  ■ Algorithms: Apriori, FP-Growth, association rule generation, GPS
  ■ Optimizations: partitioning, vertical data format, various pruning techniques

Readings

◆Textbook Chapter 6