Outline

- Association Rule Mining – Basic Concepts
- Association Rule Mining Algorithms:
  - Single-dimensional Boolean associations
  - Multi-level associations
  - Multi-dimensional associations
- Association vs. Correlation
- Adding constraints
- Applications/extensions of frequent pattern mining
- Summary
Multiple-level Association Rules

Why?
- Hard to find strong associations among low conceptual level data (e.g., less support counts for “skim milk” than “milk”)
- Associations among high-level data are likely to be known and uninteresting
- Easier to find interesting associations among items at multiple conceptual levels, rather than only among single level data

Approaches: uniform vs. reduced threshold

- Uniform min support
  - uses same min support threshold at all levels
  - Search is simplified (dealing with one threshold) and optimized (omitting itemsets that has an infrequent itemset child)
  - If the threshold is set too high → might miss associations at low level
  - if it is set too low → too many uninteresting associations

<table>
<thead>
<tr>
<th>Uniform Support</th>
<th>Reduced Support</th>
</tr>
</thead>
</table>
| Level 1  
min_sup = 5% | Level 1  
min_sup = 5% |
| Level 2  
min_sup = 5% | Level 2  
min_sup = 3% |

- Milk  
support = 10%  
Level 1 min_sup = 5%
- 2% Milk  
support = 6%  
Level 2 min_sup = 5%
- Skim Milk  
support = 4%  
Level 2 min_sup = 3%
Reduced Min Support

Four strategies:
1. **level-by-level**: full breadth search on every node
2. **level-cross filtering by single item**: items are examined only if parents are frequent (e.g., do not examine 2%Milk and Skim Milk)
3. **level-cross filtering by k-itemsets**: examine only children of frequent k-itemsets (e.g., the 2-itemset Milk&Bread is infrequent so do not examine all its children)

Top level: \( \text{min}_\text{sup} = 5\% \)
Bottom level: \( \text{min}_\text{sup} = 3\% \)

Search 1 is too relaxed, 3 is too limited, 2 is like 3 but less restricted because it deals with 1-item set

Reduced Min Support (cont)

4. **Controlled level-cross filtering by single item**: add level passage threshold (e.g., user slide the level passage threshold between 5% and 2% -- can do this for each concept hierarchy)

Top level: \( \text{min}_\text{sup} = 5\% \)
Bottom level: \( \text{min}_\text{sup} = 2\% \)

Method 2. could miss associations:
\( 2\% \text{Milk} \rightarrow \text{Skim Milk} \)

Top level: \( \text{min}_\text{sup} = 5\% \)
\( \text{level-passage-sup} = 4\% \)
Bottom level: \( \text{min}_\text{sup} = 2\% \)
Flexible Support Constraints

- Why flexible support constraints?
  - Real life occurrence frequencies vary greatly
    - Diamond, watch, pens in a shopping basket
  - Uniform support may not be an interesting model
- A flexible model
  - Usually, lower-level, more dimension combination, and longer pattern length ---> smaller support
  - General rules should be easy to specify and understand
  - Special items and special group of items may be specified individually and have higher priority

Multi-Level Mining

- A top-down, progressive deepening approach:
  - First mine high-level frequent items:
    - milk (15%), bread (10%)
  - Then mine their lower-level “weaker” frequent itemsets:
    - skim milk (5%), wheat bread (4%)
- Different min_support threshold across multi-levels lead to different algorithms:
  - If adopting the same min_support across multi-levels then toss t if any of t’s ancestors is infrequent.
  - If adopting reduced min_support at lower levels then examine only those descendents whose ancestor’s support is frequent/non-negligible.
Redundancy checking

- Must check if the resulting rules from multi-level association mining are redundant

E.g.,
1. Milk $\Rightarrow$ Bread [support 8%, confidence 70%]
2. Skim Milk $\Rightarrow$ Bread [support 2%, confidence 72%]

Suppose about 1/4 of milk sales are skim milk, then
Rule 1. can estimate that
Skim Milk $\Rightarrow$ Bread [support = 1/4 of 8% = 2%, confidence 70%]
This makes Rule 2. “redundant” since it’s closed to what is “expected”

Outline

- Association Rule Mining – Basic Concepts
- Association Rule Mining Algorithms:
  - Single-dimensional Boolean associations
  - Multi-level associations
  - Multi-dimensional associations
- Association vs. Correlation
- Adding constraints
- Applications/extensions of frequent pattern mining
- Summary
Multi-dimensional Associations

- Involve two or more dimensions (or predicates)
  
  Example:
  
  Single-dimensional rule:  \(\text{buys}(X, "milk") \Rightarrow \text{buys}(X, "bread")\)
  
  Multi-dimensional rule:  \(\text{age}(X, "0..10") \Rightarrow \text{income}(X, "0..2K")\)

- Two types of multi-dimensional assoc. rules:
  
  - **Inter-dimension** assoc. rules (*no repeated predicates*)
    \(\text{age}(X, "19-25") \land \text{occupation}(X,"student") \Rightarrow \text{buys}(X,"coke")\)
  
  - **Hybrid-dimension** assoc. rules (*repeated predicates*)
    \(\text{age}(X, "19-25") \land \text{buys}(X,"popcorn") \Rightarrow \text{buys}(X,"coke")\)

- Here we’ll deal with inter-dimension associations

Multi-dimension Mining

- Attribute types:
  
  - **Categorical**: finite number of values, no ordering among values
  
  - **Quantitative**: numeric, implicit ordering among values

- Techniques for mining multi-dimensional associations
  
  - Search for **frequent predicate sets** (as opposed to frequent itemsets)
  
  - Classified by **how “quantitative” attributes are treated**
    
    E.g., \(\{\text{age}, \text{occupation}, \text{buys}\}\) is a 3-predicate set
    
    Techniques can be categorized by how **age** values are treated
Multi-dimension Mining (MDM) Techniques

1. Concept-based
   - Quantitative attribute values are treated as predefined categories/ranges
   - Discretization occurs prior to mining using predefined concept hierarchies

2. Distribution-based
   - Quantitative attribute values are treated as quantities to satisfy some criteria (e.g., max confidence)
   - Discretization occurs during mining process using “bins” based on the distribution of the data

3. Distance-based
   - Quantitative attribute values are treated as quantities to capture meaning of interval data
   - Discretization occurs during mining process using the distance between data points

Concept-based MDM

- Numeric values are replaced by ranges or predefined concepts
- Two approaches depending on how data are stored:
  - Relational tables
    - Modify the Apriori to finding all frequent predicate sets
    - Finding k-predicate sets will require \( k \) or \( k+1 \) table scans.
  - Data cubes
    - Well suited since data cubes are multi-dimensional structures
    - The cells of n-D cuboid store support/confidence of n-predicate sets (cuboids represent aggregated dimensions)
    - To reduce candidates generated, apply the Apriori principle: every subset of frequent predicate set must be frequent
Distribution-based MDM

- Unlike concept-based approach, numeric attribute values are **dynamically** discretized to meet some criteria
  - Example of discretization: binning
    - Equiwidth: same interval size
    - Equidepth: same number of data points in each bin
    - Homogeneity-based: data points in each bin are uniformly distributed
  - Example of criteria:
    - Compact
    - Strong rules (i.e., high confidence/support)

- Resulting rules are referred to as **Quantitative Association Rules**
- Consider a 2-D quantitative association rule: $A_{\text{quant}1} \land B_{\text{quant}2} \Rightarrow C_{\text{cat}}$
  
  E.g., $\text{age}(X, "30-39") \land \text{income}(X, "40K-44K") \Rightarrow \text{buys}(X, "HD TV")$

### Distribution-based MDM - Example

**ARCS – Association Rule Clustering System**

- For each quantitative attribute, discretize the numeric values based on the data distribution, e.g., by binning techniques
  - 2-D table of the resulting bins of the two quantitative attributes on LHS of the rule
  - Each cell holds count distribution in each category of the attribute on the RHS of the rule

- Finding frequent predicate sets
  - Generate strong associations (same as in Apriori)
    - age(X,"30-34") $\land$ income(X,"40K-44K") $\Rightarrow$ buys(X,"HD TV")
    - age(X,"35-39") $\land$ income(X,"40K-44K") $\Rightarrow$ buys(X,"HD TV")

- Simplify resulting rules
  - Rule “clusters” (here in “grids”) are further combined
    - age(X,"30-39") $\land$ income(X,"40K-44K") $\Rightarrow$ buys(X,"HD TV")
Distance-based MDM

- Binning methods do not capture the semantics of interval data, e.g., Price ($): 7 20 22 50 51 53

<table>
<thead>
<tr>
<th>Equi-width (width $10)</th>
<th>Equi-depth (depth 2)</th>
<th>Distance-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0,10]</td>
<td>[7,20]</td>
<td>[7,7]</td>
</tr>
<tr>
<td>[11,20]</td>
<td>[22,50]</td>
<td>[20,22]</td>
</tr>
<tr>
<td>[21,30]</td>
<td>[51,53]</td>
<td>[50,53]</td>
</tr>
<tr>
<td>[31,40]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[41,50]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[51,60]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Distance-based partitioning, more meaningful discretization considering:
  - density/number of points in an interval
  - “closeness” of points in an interval

Distance-based MDM (contd)

- Distance measures: e.g., two points \((x_1,x_2,x_3)\) and \((t_1,t_2,t_3)\)
  - Euclidean \(\sqrt{\sum_{i=1}^{3}(x_i-t_i)^2}\)
  - Manhattan \(\sum_{i=1}^{3}|x_i-t_i|\)
- Two phases: \(\sum_{i=1}^{3}|x_i-t_i|\)
  - Identify clusters (Ch 8)
    - Data points in each cluster satisfy both frequency threshold and density threshold ~ support
  - Obtain association rules
    - Define degree of associations ~ confidence, e.g., centroid (average of data points in the cluster) Manhattan distance
- Three conditions:
  - Clusters in LHS are each strongly associated with each clusters in RHS
  - Clusters in LHS collectively occur together
  - Clusters in RHS collectively occur together
Outline

- Association Rule Mining – Basic Concepts
- Association Rule Mining Algorithms:
  - Single-dimensional Boolean associations
  - Multi-level associations
  - Multi-dimensional associations
- Association vs. Correlation
- Adding constraints
- Applications/extensions of frequent pattern mining
- Summary

Association & Correlation analysis

<table>
<thead>
<tr>
<th></th>
<th>Basketball</th>
<th>Not basketball</th>
<th>Sum (row)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cereal</td>
<td>400</td>
<td>350</td>
<td>750</td>
</tr>
<tr>
<td>Not cereal</td>
<td>200</td>
<td>50</td>
<td>250</td>
</tr>
<tr>
<td>Sum(col.)</td>
<td>600</td>
<td>400</td>
<td>1000</td>
</tr>
</tbody>
</table>

- Suppose: min support 20%, min confidence = 50%
- Probability of buying cereal = 750/1000 = 75%
- basketball ⇒ cereal [400/1000 = 40%, 400/600 = 66.7%]
  - Chance of buying cereal (even without this rule) is already higher than 66.7%
  - the implication of this rule is not interesting
    - “strong” rule (high conf) but “uninformative” (prob on RHS > conf)
Association & Correlation analysis (contd)

<table>
<thead>
<tr>
<th></th>
<th>Basketball</th>
<th>Not basketball</th>
<th>Sum (row)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cereal</td>
<td>400</td>
<td>350</td>
<td>750</td>
</tr>
<tr>
<td>Not cereal</td>
<td>200</td>
<td>50</td>
<td>250</td>
</tr>
<tr>
<td>Sum(col.)</td>
<td>600</td>
<td>400</td>
<td>1000</td>
</tr>
</tbody>
</table>

- Define $corr_{A,B} = \frac{P(A \cap B)}{P(A)P(B)} = \begin{cases} 1 & \text{if } A \text{ and } B \text{ are independent} \\ < 1 & \text{if } A \text{ & } B \text{ are -ve correlated} \\ > 1 & \text{if } A \text{ & } B \text{ are +ve correlated} \end{cases}$
  
  $= \text{Lift}(A \Rightarrow B)$

- $corr_{\text{basketball, cereal}} = \frac{(400/1000)}{(600/1000)(750/1000)} = 0.89$
  
  $\Rightarrow$ basketball and cereal are negatively correlated

- $corr_{\text{basketball, not cereal}} = \frac{(200/1000)}{(600/1000)(250/1000)} = 1.3$
  
  $\Rightarrow$ basketball and not cereal are positively correlated

But $\text{basketball} \Rightarrow \text{not cereal}$ [200/1000 = 20%, 200/600 = 33.3%]

“Not strong” but “informative” (prob of not buying cereal only 25%)

Association & Correlation analysis (contd)

- Association and Correlation are not the same
  
  - $\text{basketball} \Rightarrow \text{cereal}$
    
    strong
    
    uninformative & -vely correlated
  
  - $\text{basketball} \Rightarrow \text{not cereal}$
    
    not strong
    
    informative & +vely correlated

- $-ve \ corr$: $P(A \& B) < P(A) P(B)$

- $Informative$: $P(B) < \frac{\text{Conf}(A \Rightarrow B)}{P(A)}$

- $P(B) < \frac{P(A \& B)}{P(A)}$

- $P(B) P(A) < P(A \& B)$

- $-ve \ corr = \text{uninformative}$

Can LHS and RHS of a rule be negatively correlated and yet the rule is informative?
Association & Correlation analysis (contd)

- Association and Correlation are not the same
- Mining of correlated rules
  - I.e., rules involve correlated itemsets (instead of frequent itemsets)
  - Correlation value of a set of items can be calculated (cf. $corr_{A,B}$)
  - Use the $\chi^2$ statistic to test if the correlation value is statistically significant
  - **Upward closed property** – If $A$ has a property, so is $A'$'s superset
    - Correlation is upward closed ($A$ is a correlated itemset, so is its superset)
    - $\chi^2$ is upward closed (within each significance level)
  - Search upward for correlated itemsets starting from an empty set to find minimal correlated item sets
    - In datacube – random walk algorithms are used
    - In general – still an open problem when dealing with large dimensions
- See also [Brin et al., 97]

Outline

- Association Rule Mining – Basic Concepts
- Association Rule Mining Algorithms:
  - Single-dimensional Boolean associations
  - Multi-level associations
  - Multi-dimensional associations
- Association vs. Correlation
- Adding constraints
- Applications/extensions of frequent pattern mining
- Summary
Constraint-based Mining

- Finding all the patterns in a database autonomously? — unrealistic!
  - The patterns could be too many but not focused!

- Constraint-based mining allows
  - Specification of constraints on what to be mined → more effective mining, e.g.,
    - **Metarule**: Template $A(x,y) + B(x,w) \Rightarrow buys(x, \text{"HD TV"})$ to guide search
    - **Rule constraint**: small sales (price<$10) triggers big sales (sum>$200)

- System optimization → more efficient mining, e.g., data mining query optimization

- Constraint-based mining aims to reduce search and find all answers that satisfy a given constraint

Constrained Frequent Pattern Mining

A Mining Query Optimization Problem

- Given a frequent pattern mining query with a set of constraints $C$, the algorithm should be
  - **sound**: it only finds frequent sets that satisfy $C$
  - **complete**: all frequent sets satisfying $C$ are found

- A naïve solution:
  - First find all frequent sets, and then test them for constraint satisfaction

- More efficient approaches:
  - Analyze the properties of constraints comprehensively
  - **Push them as deeply as possible inside** the frequent pattern computation and still ensure completeness of the answer.

**Which is harder?**

**What kind of rule constraints can be pushed as above?**
Rule constraints

- Types of rule constraints:
  - Anti-monotone
  - Monotone
  - Succinct
  - Convertible
  - Inconvertible
- The first four types can be pushed in the mining process to improve efficiency without losing completeness of the answers.

(Anti-)monotone constraints

- \( c \) = a rule constraint
- \( A \) = an itemset, \( B \) = a proper superset of \( A \)
- **Monotone**: \( A \) satisfies \( c \) \( \Rightarrow \) any \( B \) satisfies \( c \)
- **Anti-monotone**: \( A \) doesn’t satisfy \( c \) \( \Rightarrow \) none of \( B \) satisfies \( c \)

Examples:
- \( \text{sum}(A.Price) \geq v \) is monotone
- \( \text{min}(A.Price) \leq v \) is monotone
- \( \text{sum}(A.Price) \leq v \) is anti-monotone
- \( \text{min}(A.Price) \geq v \) is anti-monotone
- \( C: \text{range}(A.profit) \leq 15 \) is anti-monotone
  - Itemset \( ab \) violates \( C \)
  - So does every superset of \( ab \)

<table>
<thead>
<tr>
<th>Item</th>
<th>Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>40</td>
</tr>
<tr>
<td>b</td>
<td>0</td>
</tr>
<tr>
<td>c</td>
<td>-20</td>
</tr>
<tr>
<td>d</td>
<td>10</td>
</tr>
<tr>
<td>e</td>
<td>-30</td>
</tr>
<tr>
<td>f</td>
<td>30</td>
</tr>
<tr>
<td>g</td>
<td>20</td>
</tr>
<tr>
<td>h</td>
<td>-10</td>
</tr>
</tbody>
</table>
Succinct constraints

- **Succinct**: there is a “formula” to generate precisely all itemsets satisfying the constraint
  - itemsets satisfying the constraint can be enumerated before support counting starts
  - Succinct constraints are *pre-counting prunable*

Examples:
- \( c: \ max(A.Price) \geq 20 \) is monotone and succinct
  - An itemset satisfies \( c \) is of the form \( A_1 \cup A_2 \), where
    - \( A_2 \) is \( \{b\} - \) a set (can be empty) of items with prices \( \leq v \)
    - \( A_1 \) is a non-empty subset of \( \{a, c, d, e\} \) - a set of items with prices \( \geq v \)
- \( \min(A.Price) \leq v \) is succinct and monotone
- \( \sum(A.Price) \leq v \) is not succinct but anti-monotone
- \( \sum(A.Price) \geq v \) is not succinct but monotone

<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>40</td>
</tr>
<tr>
<td>b</td>
<td>10</td>
</tr>
<tr>
<td>c</td>
<td>22</td>
</tr>
<tr>
<td>d</td>
<td>25</td>
</tr>
<tr>
<td>e</td>
<td>30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TID</th>
<th>Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a, b, c, d</td>
</tr>
<tr>
<td>20</td>
<td>a, c, d</td>
</tr>
<tr>
<td>30</td>
<td>a, b, d</td>
</tr>
</tbody>
</table>

The Apriori Algorithm — Example

Database D

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1 3 4</td>
</tr>
<tr>
<td>200</td>
<td>2 3 5</td>
</tr>
<tr>
<td>300</td>
<td>1 2 3 5</td>
</tr>
<tr>
<td>400</td>
<td>2 5</td>
</tr>
</tbody>
</table>

Scan D

\( C_1 \)

\[ \begin{align*}
&\{1\} & 2 \\
&\{1\} & 3 \\
&\{3\} & 3 \\
&\{4\} & 1 \\
&\{5\} & 3 \\
\end{align*} \]

\( L_1 \)

\[ \begin{align*}
&\{1\} & 2 \\
&\{2\} & 3 \\
&\{3\} & 3 \\
&\{5\} & 3 \\
\end{align*} \]

Scan D

\( C_2 \)

\[ \begin{align*}
&\{1\} & 1 \\
&\{1\} & 2 \\
&\{1\} & 3 \\
&\{5\} & 1 \\
\end{align*} \]

\( L_2 \)

\[ \begin{align*}
&\{1\} & 2 \\
&\{1\} & 2 \\
&\{2\} & 2 \\
&\{3\} & 2 \\
&\{3\} & 3 \\
\end{align*} \]

Scan D

\( C_3 \)

\[ \begin{align*}
&\{1\} & 1 \\
&\{2\} & 2 \\
&\{3\} & 2 \\
&\{5\} & 2 \\
\end{align*} \]

\( L_3 \)

\[ \begin{align*}
&\{2\} & 1 \\
&\{2\} & 1 \\
&\{2\} & 2 \\
&\{3\} & 3 \\
\end{align*} \]
### Naïve: Apriori + Constraint:

\[ \text{Sum}(S, \text{price} < 5) \]

**price of item \( k \) is \( k \)**

<table>
<thead>
<tr>
<th>Database D</th>
<th>Itemset</th>
<th>Sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>TID</td>
<td>Items</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>1 3 4</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>2 3 5</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>1 2 3 5</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>2 5</td>
<td></td>
</tr>
</tbody>
</table>

**Scan D**

<table>
<thead>
<tr>
<th>( C_1 )</th>
<th>Itemset</th>
<th>Sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>(2)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>(3)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>(4)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(5)</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**Scan D**

<table>
<thead>
<tr>
<th>( C_2 )</th>
<th>Itemset</th>
<th>Sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1 2)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(1 3)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>(1 5)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(2 3)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>(2 5)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>(3 5)</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**Scan D**

<table>
<thead>
<tr>
<th>( C_3 )</th>
<th>Itemset</th>
<th>Sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2 3 5)</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

### Pushing constraint:

\[ \text{Sum}(S, \text{price} < 5) \]

**price of item \( k \) is \( k \)**

<table>
<thead>
<tr>
<th>Database D</th>
<th>Itemset</th>
<th>Sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>TID</td>
<td>Items</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>1 3 4</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>2 3 5</td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>1 2 3 5</td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>2 5</td>
<td></td>
</tr>
</tbody>
</table>

**Scan D**

<table>
<thead>
<tr>
<th>( C_1 )</th>
<th>Itemset</th>
<th>Sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>(2)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>(3)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>(4)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(5)</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

**Scan D**

<table>
<thead>
<tr>
<th>( C_2 )</th>
<th>Itemset</th>
<th>Sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1 2)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(1 3)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>(1 5)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>(2 3)</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>(2 5)</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>(3 5)</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>

**Scan D**

<table>
<thead>
<tr>
<th>( C_3 )</th>
<th>Itemset</th>
<th>Sup</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2 3 5)</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
Pushing Succinct Constraint: \( \text{Min}\{S.\text{price} \leq 1\} \)

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>1 3 4</td>
</tr>
<tr>
<td>200</td>
<td>2 3 5</td>
</tr>
<tr>
<td>300</td>
<td>1 2 3 5</td>
</tr>
<tr>
<td>400</td>
<td>2 5</td>
</tr>
</tbody>
</table>

Database D

Scan D

\( L_1 \)

\( L_2 \)

\( L_3 \)

Convertible constraints

- Constraints that can become anti-monotone or monotone when items in itemsets are ordered in a certain way.

Example:

\( C: \text{avg}(S.\text{profit}) \geq 15 \)

\( C \) is not anti-monotone nor monotone

If items are added in value-descending order:

\(<a, f, g, d, b, h, c, e>\)

\( gb \) violates \( C \), so does \( gbh \), and \( gb^* \) (note * = strings representing itemsets with each item value \( \leq b \)'s value)

\( C \) becomes anti-monotone

\( C \) with respect to value-descending order is anti-monotone convertible.
Strongly Convertible Constraints

- $\text{avg}(X) \geq 15$ is convertible anti-monotone w.r.t. item value descending order $R: <a, f, g, d, b, h, c, e>$

- $\text{avg}(X) \geq 15$ is convertible monotone w.r.t. item value ascending order $R^{-1}: <e, c, h, b, d, g, f, a>$

- We say, $\text{avg}(X) \geq 15$ is strongly convertible

More examples

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Convertible anti-monotone</th>
<th>Convertible monotone</th>
<th>Strongly convertible</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{avg}(S) \leq v, \geq v$</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$\text{median}(S) \leq v$</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>$\text{sum}(S) \leq v$ (items could be of any value, $v \geq 0$)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>$\text{sum}(S) \leq v$ (items could be of any value, $v \leq 0$)</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>$\text{sum}(S) \geq v$ (items could be of any value, $v \geq 0$)</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>$\text{sum}(S) \geq v$ (items could be of any value, $v \leq 0$)</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>......</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Common SQL-based constraints

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Antimonotone</th>
<th>Monotone</th>
<th>Succinct</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v \in S )</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>( S \subseteq V )</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>( S \subseteq V )</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>min(S) ( \leq v )</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>min(S) ( \geq v )</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>max(S) ( \leq v )</td>
<td>yes</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>max(S) ( \geq v )</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
</tr>
<tr>
<td>count(S) ( \leq v )</td>
<td>yes</td>
<td>no</td>
<td>weakly</td>
</tr>
<tr>
<td>count(S) ( \geq v )</td>
<td>yes</td>
<td>no</td>
<td>weakly</td>
</tr>
<tr>
<td>sum(S) ( \leq v ) ( a ( \in S, a \geq 0 ) )</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>sum(S) ( \geq v ) ( a ( \in S, a \geq 0 ) )</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>range(S) ( \leq v )</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>range(S) ( \geq v )</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>avg(S) ( \theta v, \theta \in {=, \leq, \geq} )</td>
<td>convertible</td>
<td>convertible</td>
<td>no</td>
</tr>
<tr>
<td>support(S) ( \geq \xi )</td>
<td>yes</td>
<td>no</td>
<td>no</td>
</tr>
<tr>
<td>support(S) ( \leq \xi )</td>
<td>no</td>
<td>yes</td>
<td>no</td>
</tr>
</tbody>
</table>

Classification of Constraints
Mining with convertible constraints

- $C$: $\text{avg}(S\text{.profit}) \geq 25$
- List items in every transaction in value descending order $R$: $<a, f, g, d, b, h, c, e>$
  - $C$ is convertible anti-monotone w.r.t. $R$
- Scan transaction DB once
  - remove infrequent items: drop $h$
- $C$ can’t be pushed in level-wise framework
  - Itemset $df$ violates $C$ - we want to prune it
  - Since $adf$ satisfies $C$, Apriori needs $df$ to assemble $adf$, $df$ cannot be pruned
- But $C$ can be pushed into frequent-pattern growth framework!

<table>
<thead>
<tr>
<th>TID</th>
<th>Transaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>a, b, c, d, f</td>
</tr>
<tr>
<td>20</td>
<td>b, c, d, f, g</td>
</tr>
<tr>
<td>30</td>
<td>a, c, d, e, f</td>
</tr>
<tr>
<td>40</td>
<td>c, e, f, g, h</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Item</th>
<th>sup</th>
<th>Profit</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>4</td>
<td>40</td>
</tr>
<tr>
<td>f</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>2</td>
<td>20</td>
</tr>
<tr>
<td>d</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>b</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>h</td>
<td>1</td>
<td>-10</td>
</tr>
<tr>
<td>c</td>
<td>4</td>
<td>-20</td>
</tr>
<tr>
<td>e</td>
<td>2</td>
<td>-30</td>
</tr>
</tbody>
</table>

Recap: Constraint-based mining

- All types of rule constraints but inconvertible can be used to guide the mining process to improve mining efficiency
- Anti-monotone constraints can be applied at each iteration of Apriori-like algorithms while guaranteeing completeness
  - Pushing non-anti-monotone constraints into the mining process will not guarantee completeness
- Itemsets satisfy succinct constraints can be determined before support counting begins
  - no need to iteratively check the rule constraint during the mining process
  - succinct constraints are pre-computing pushable
- Convertible constraints can’t be pushed in level-wise mining algorithm such as Apriori
Handling Multiple Constraints

- Different constraints may require different or even conflicting item-ordering
- If there exists an order $R$ s.t. both $C_1$ and $C_2$ are convertible w.r.t. $R$, then there is no conflict between the two convertible constraints
- If there exists conflict on order of items
  - Try to satisfy one constraint first
  - Then using the order for the other constraint to mine frequent itemsets in the corresponding projected database

Outline

- Association Rule Mining – Basic Concepts
- Association Rule Mining Algorithms:
  - Single-dimensional Boolean associations
  - Multi-level associations
  - Multi-dimensional associations
- Association vs. Correlation
- Adding constraints
- Applications/extensions of frequent pattern mining
- Summary
Extensions/applications

- The following is not an exhaustive list
- Some topics are likely to be assigned for your presentations in the second half of this class

Sequential Pattern Mining

- **Sequence data vs. Time-series data**
  - sequences of ordered events (with or without explicit notion of time)
  - sequences of values/events typically measured at equal time intervals
- **Time-series data are sequence data but not viz.**
- **Sequential Pattern mining**
  - Deals with frequent sequential patterns (as opposed to frequent patterns)
  - Problem: given a set of sequences, find the complete set of frequent subsequences
- **Applications of sequential pattern mining**
  - Customer shopping sequences, e.g., First buy computer, then CD-ROM, and then digital camera, within 3 months.
  - Medical treatment, natural disasters (e.g., earthquakes), science & engineering processes, stocks and markets, etc.
  - Telephone calling patterns, Weblog click streams
  - DNA sequences and gene structures
Studies on Sequential Pattern Mining

- Concept introduction and an initial Apriori-like algorithm
- GSP—An Apriori-based, influential mining method (developed at IBM Almaden)
- From sequential patterns to episodes (Apriori-like + constraints)
- Mining sequential patterns with constraints

Classification-Based on Associations

- Mine association possible rules (PR) in form of condset \( \rightarrow \) c
  - Condset: a set of attribute-value pairs
  - C: class label
- Build Classifier
  - Organize rules according to decreasing precedence based on confidence and support
Iceberg Cube computation

- It is too costly to materialize a high dimension cube
  - 20 dimensions each with 99 distinct values may lead to $100^{20}$ cube cells
  - Even if there is only one nonempty cell in each $10^{10}$ cells, the cube will still contain $10^{30}$ nonempty cells

- Observation: Trivial cells are usually not interesting
  - Nontrivial: large volume of sales, or high profit

- Solution:
  - Iceberg cube—materialize only nontrivial cells of a data cube – cf. tip of the iceberg
  - Computation: Based on Apriori-like pruning, e.g.,
    - BUC [Bayer & Ramakrishnan, 99]
    - bottom-up cubing, efficient bucket-sort alg.
    - Only handles anti-monotonic iceberg cubes
      - If a cell $c$ violates the HAVING clause, so do all more specific cells

Spatial and Multi-Media Association

A Progressive Refinement Method: Why?

- Mining operator can be expensive or cheap, fine or rough
- Superset coverage property:
  - Preserve all the positive answers—allow a positive false test but not a false negative test.
- Two- or multi-step mining:
  - First apply rough/cheap operator (superset coverage)
  - Then apply expensive algorithm on a substantially reduced candidate set (Koperski & Han, SSD’ 95).
Spatial Associations

- Hierarchy of spatial relationship:
  - “g_close_to”: near_by, touch, intersect, contain, etc.
  - First search for rough relationship and then refine it.

- Two-step mining of spatial association:
  - Step 1: rough spatial computation (as a filter)
  - Step 2: Detailed spatial algorithm (as refinement)
    - Apply only to those objects which have passed the rough spatial association test (no less than min_support)

Mining Multimedia Associations

Correlations with color, spatial relationships, etc. From coarse to fine resolution mining
Outline

- Association Rule Mining – Basic Concepts
- Association Rule Mining Algorithms:
  - Single-dimensional Boolean associations
  - Multi-level associations
  - Multi-dimensional associations
- Association vs. Correlation
- Adding constraints
- Applications/extensions of frequent pattern mining
- Summary

Achievements

- Frequent pattern mining—an important task in data mining
- Frequent pattern mining methodology
  - Candidate generation-test vs. projection-based (frequent-pattern growth)
  - Vertical vs. horizontal format (itemsets vs. transactionsets)
  - Various optimization methods: database partition, scan reduction, hash tree, sampling, border computation, clustering, etc.
- Related frequent pattern mining algorithm: scope extension
  - Mining closed frequent itemsets and max-patterns (e.g., MaxMiner, CLOSET, CHARM, etc.)
  - Mining multi-level, multi-dimensional frequent patterns with flexible support constraints
  - Constraint pushing for mining optimization
  - From frequent patterns to correlation and causality
Applications

- Related problems which need frequent pattern mining
  - Association-based classification
  - Iceberg cube computation
  - Database compression by frequent patterns
  - Mining sequential patterns (GSP, PrefixSpan, SPADE, etc.)
    - Mining partial periodicity, cyclic associations, etc.
    - Mining frequent structures, trends, etc.
- Typical application examples
  - Market-basket analysis, Weblog analysis, DNA mining, etc.

Some Research Problems

- Multi-dimensional gradient analysis: patterns regarding changes and differences
  - Not just counts—other measures, e.g., avg(profit)
- Mining top-k frequent patterns without support constraint
- Partial periodic patterns
- DNA sequence analysis and pattern classification
References

Frequent-pattern Mining Methods

- J. Han, J. Pei, and Y. Yin; “Mining frequent patterns without candidate generation”. In Proc. ACM-SIGMOD’ 2000, pp. 1-12, Dallas, TX, May 2000.

- R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. SIGMOD’96, 1-12, Montreal, Canada.
References

Performance Improvements

- D.W. Cheung, J. Han, V. Ng, and C.Y. Wong. Maintenance of discovered association rules in large databases: An incremental updating technique. ICDE’96, New Orleans, LA.
- E.-H. Han, G. Karypis, and V. Kumar. Scalable parallel data mining for association rules. SIGMOD’97, Tucson, Arizona.

References

Performance Improvements

- J.S. Park, M.S. Chen, and P.S. Yu. An effective hash-based algorithm for mining association rules. SIGMOD’95, San Jose, CA.
References

Multi-level, correlation, ratio rules, etc

- J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. VLDB'95, 420-431, Zurich, Switzerland.
- J. Pei, A.K.H. Tung, J. Han. Fault-Tolerant Frequent Pattern Mining: Problems and Challenges. SIGMOD DMKD’01, Santa Barbara, CA.

Mining Max-patterns and Closed itemsets

- J. Pei, J. Han, and R. Mao, "CLOSEST: An Efficient Algorithm for Mining Frequent Closed Itemsets", Proc. 2000 ACM-SIGMOD Int. Workshop on Data Mining and Knowledge Discovery (DMKD'00), Dallas, TX, May 2000.
- M. Zaki. CHARM: An Efficient Algorithm for Closed Association Rule Mining, SIAM’ 02
References

Constraint-based Mining

- J. Han, L. V. S. Lakshmanan, and R. T. Ng, "Constraint-Based, Multidimensional Data Mining", COMPUTER (special issues on Data Mining), 32(8): 46-50, 1999.
- L. V. S. Lakshmanan, R. Ng, J. Han, and A. Pang, "Optimization of Constrained Frequent Set Queries with 2-Variable Constraints", SIGMOD'99.
- R. Ng, L.V.S. Lakshmanan, J. Han, and A. Pang, "Exploratory mining and pruning optimizations of constrained association rules.", SIGMOD'98.
- J. Pei, J. Han, and L. V. S. Lakshmanan, "Mining Frequent Itemsets with Convertible Constraints", Proc. 2001 Int. Conf. on Data Engineering (ICDE'01), April 2001.
- J. Pei and J. Han "Can We Push More Constraints into Frequent Pattern Mining?", Proc. 2000 Int. Conf. on Knowledge Discovery and Data Mining (KDD'00), Boston, MA, August 2000.

Sequential Pattern Mining Methods

- R. Agrawal and R. Srikant. Mining sequential patterns. ICDE'95, 3-14, Taipei, Taiwan.
- J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, M.-C. Hsu, "FreeSpan: Frequent Pattern-Projected Sequential Pattern Mining", Proc. 2000 Int. Conf. on Knowledge Discovery and Data Mining (KDD'00), Boston, MA, August 2000.
- J. Pei, J. Han, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth", Proc. 2001 Int. Conf. on Data Engineering (ICDE'01), Heidelberg, Germany, April 2001.
- B. Ozden, S. Ramaswamy, and A. Silberschatz. Cyclic association rules. ICDE'98, 412-421, Orlando, FL.
References

Mining in Spatial, Multimedia, Text & Web Databases

- A. K. H. Tung, H. Lu, J. Han, and L. Feng, "Breaking the Barrier of Transactions: Mining Inter-Transaction Association Rules", Proc. 1999 Int. Conf. on Knowledge Discovery and Data Mining (KDD'99), San Diego, CA, Aug. 1999, pp. 297-301.
- J. Han, G. Dong and Y. Yin, "Efficient Mining of Partial Periodic Patterns in Time Series Database", Proc. 1999 Int. Conf. on Data Engineering (ICDE'99), Sydney, Australia, March 1999, pp. 106-115
- O. R. Zaiane, M. Xin, J. Han, "Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs," Proc. Advances in Digital Libraries Conf. (ADL'98), Santa Barbara, CA, April 1998, pp. 19-29

Mining for Classification and Data Cube Computation

- M. Kamber, J. Han, and J. Y. Chiang. Metarule-guided mining of multi-dimensional association rules using data cubes. KDD'97, 207-210, Newport Beach, California.