Today’s Reading:
- HMS 6.7, ch. 13

Today’s Lecture:
- Finding Patterns and Rules
  - Association Rules

Upcoming Due Dates:
- H3 due today
- Project Writeup due 4/30
Big Picture

- **Predictive Models**: emphasis on a single attribute
- **Pattern Discovery**: characterizes local aspects of data
- **Descriptive Models**: full description of the data
Finding Patterns and Rules

• Examples:
  – supermarket transaction database,
    *10% of the customers buy wine and cheese*

  – telecommunications alarms database
    *if alarms A and B occur within 30 seconds of each other, then alarm C occurs within 60 seconds with probability 0.5*

  – web log dataset
    *if a person visits the CNN Web site, there is 60% chance the person will visit the ABC News Web site in the same month*
Mining Rules

• Problems?
  – number of rules grows exponentially with the number of attributes
  – among the large number of rules generated, how do we determine interesting rules?
What Is Association Mining?

- **Association rule mining:**
  - Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.

- **Applications:**
  - Basket data analysis, cross-marketing, catalog design, loss-leader analysis, clustering, classification, etc.

- **Examples.**
  - Rule form: “Body $\rightarrow$ Head [support, confidence]”.
  - `b`uys(x, “diapers”) $\rightarrow$ buys(x, “beers”) [0.5%, 60%]
  - major(x, “CS”) $\wedge$ takes(x, “DB”) $\rightarrow$ grade(x, “A”) [1%, 75%]
Association Rule Flavors

- **Boolean vs. quantitative associations** *(Based on the types of values handled)*
  - \( \text{buys}(x, \text{"SQLServer"}) \land \text{buys}(x, \text{"DMBook"}) \rightarrow \text{buys}(x, \text{"DBMiner"}) \) [0.2%, 60%]
  - \( \text{age}(x, \text{"30..39"}) \land \text{income}(x, \text{"42..48K"}) \rightarrow \text{buys}(x, \text{"PC"}) \) [1%, 75%]

- **Single dimension vs. multiple dimensional associations**

- **Single level vs. multiple-level analysis**
  - What brands of beers are associated with what brands of diapers?

- **Various extensions**
  - Correlation, causality analysis
    - Association does not necessarily imply correlation or causality
  - Maxpatterns and closed itemsets
  - Constraints enforced
    - E.g., small sales (sum < 100) trigger big buys (sum > 1,000)?
Market Basket Analysis

• Given some set of *items*
  1. database of transactions,
  2. each transaction is a list of items (purchased by a customer in a visit), called a *basket*

• Find: all rules that correlate the presence of one set of items with that of another set of items in a basket
  - E.g., *98% of people who purchase tires and auto accessories also get automotive services done*

• Other problems with this structure:
  - baskets = documents; items = words
  - baskets = web pages; items = links
Market Basket Analysis, cont

• Typically transaction database too large to fit in main memory
  – common sizes
    number of transactions: $10^5 - 10^8$
    number of items: $10^2 - 10^6$
  – typically sparse, 0.1% chance of random purchase
  – commonly stored
    • in a relational data base, Baskets(BID,item)
    • flat file (BID,item_1,item_2,...,item_n)

• Evaluating running time:
  – count the number of passes through the data
  – principle cost time to read data from disk
Rule Measures: Support and Confidence

Find all the rules $X \& Y \Rightarrow Z$ with minimum confidence and support

- **support**, $s$, proportion of transactions containing $\{X, Y, Z\}$
- **confidence**, $c$, proportion of transactions which have $\{X, Y\}$ and also contain $Z$
Example: Support and Confidence

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Items Bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>A,B,C</td>
</tr>
<tr>
<td>1000</td>
<td>A,C</td>
</tr>
<tr>
<td>4000</td>
<td>A,D</td>
</tr>
<tr>
<td>5000</td>
<td>B,E,F</td>
</tr>
</tbody>
</table>

Let minimum support 50%, and minimum confidence 50%, we have:

\[ A \Rightarrow C \quad S = 50\% \quad C = 66.6\% \]

\[ C \Rightarrow A \quad S = 50\% \quad C = 100\% \]
Mining Association Rules—An Example

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Items Bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>2000</td>
<td>A, B, C</td>
</tr>
<tr>
<td>1000</td>
<td>A, C</td>
</tr>
<tr>
<td>4000</td>
<td>A, D</td>
</tr>
<tr>
<td>5000</td>
<td>B, E, F</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frequent Itemset</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>{A}</td>
<td>75%</td>
</tr>
<tr>
<td>{B}</td>
<td>50%</td>
</tr>
<tr>
<td>{C}</td>
<td>50%</td>
</tr>
<tr>
<td>{A, C}</td>
<td>50%</td>
</tr>
</tbody>
</table>

For rule $A \Rightarrow C$:
- support = support({$A, C$}) = 50%
- confidence = support({$A, C$})/support({$A$}) = 66.6%

The **Apriori** principle:
Any subset of a frequent itemset must be frequent
Finding Frequent Itemsets

• Find the *frequent itemsets*: the sets of items that have minimum support
  
  – **Monotonicity principle**: A subset of a frequent itemset must also be a frequent itemset
    
    • i.e., if \{AB\} is a frequent itemset, both \{A\} and \{B\} should be a frequent itemset
    
    – Iteratively find frequent itemsets with cardinality from 1 to \(k\) (\(k\)-itemset)

• Use the frequent itemsets to generate association rules.
Finding frequent itemsets, cont.

- **Two approaches**
  - Proceed levelwise, first find frequent items (sets of size 1), then frequent pairs, next frequent triples
    - most time consuming step is often finding pairs
    - one pass over the data is made for each level
  - Find all maximal frequent itemsets (I.e., sets S such that no proper superset of S is frequent)
    - one or several passes over the data
The Apriori Algorithm

- **Join Step**: $C_k$ is generated by joining $L_{k-1}$ with itself.
- **Prune Step**: Any $(k-1)$-itemset that is not frequent cannot be a subset of a frequent $k$-itemset.

$C_k$: Candidate itemset of size $k$
$L_k$: frequent itemset of size $k$

$L_1 = \{\text{frequent items}\}$;

\[
\text{for } (k = 1; L_k \neq \emptyset; k++) \text{ do begin}
\]
- $C_{k+1} = \text{candidates generated from } L_k$;
- \textbf{for each} transaction $t$ in database do
  - increment the count of all candidates in $C_{k+1}$ that are contained in $t$
- $L_{k+1} = \text{candidates in } C_{k+1} \text{ with min\_support}$
\[
\text{end}
\]

\textbf{return } \bigcup_k L_k ;
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

\[ C_1 \]

itemset | sup.
--- | ---
{1} | 2
{2} | 3
{3} | 3
{4} | 1
{5} | 3

itemset sup.

\[ L_1 \]

{1} | 2
{2} | 3
{3} | 3
{5} | 3

\[ C_2 \]

itemset sup.

\[ L_2 \]

itemset sup.

\[ C_3 \]

{2 3 5} | 2

Scan D

\[ L_3 \]

{2 3 5} | 2
How to Generate Candidates?

• Suppose the items in $L_{k-1}$ are ordered

• Step 1: self-joining $L_{k-1}$
  
  insert into $C_k$
  
  select $p.item_1, p.item_2, ..., p.item_{k-1}, q.item_{k-1}$
  
  from $L_{k-1} p, L_{k-1} q$
  
  where $p.item_1=q.item_1, ..., p.item_{k-2}=q.item_{k-2}, p.item_{k-1} < q.item_{k-1}$

• Step 2: pruning

  forall itemsets $c$ in $C_k$ do
  
  forall $(k-1)$-subsets $s$ of $c$ do
  
  if ($s$ is not in $L_{k-1}$) then delete $c$ from $C_k$
How to Count Supports of Candidates?

- Why counting supports of candidates a problem?
  - The total number of candidates can be very huge
  - One transaction may contain many candidates

- Method:
  - Candidate itemsets are stored in a *hash-tree*
  - *Leaf node* of hash-tree contains a list of itemsets and counts
  - *Interior node* contains a hash table
  - *Subset function*: finds all the candidates contained in a transaction
Example of Generating Candidates

- $L_3 = \{abc, abd, acd, ace, bcd\}$
- Self-joining: $L_3 \times L_3$
  - $abcd$ from $abc$ and $abd$
  - $acde$ from $acd$ and $ace$
- Pruning:
  - $acde$ is removed because $ade$ is not in $L_3$
- $C_4 = \{abcd\}$
Methods to Improve Apriori’s Efficiency

- **Hash-based itemset counting**: A $k$-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent.
- **Transaction reduction**: A transaction that does not contain any frequent $k$-itemset is useless in subsequent scans.
- **Partitioning**: Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB.
- **Sampling**: mining on a subset of given data, lower support threshold + a method to determine the completeness.
- **Dynamic itemset counting**: add new candidate itemsets only when all of their subsets are estimated to be frequent.
Is Apriori Fast Enough? — Performance Bottlenecks

• The core of the Apriori algorithm:
  – Use frequent \((k-1)\)-itemsets to generate candidate frequent \(k\)-itemsets
  – Use database scan and pattern matching to collect counts for the candidate itemsets

• The bottleneck of Apriori: candidate generation
  – Huge candidate sets:
    • \(10^4\) frequent 1-itemset will generate \(10^7\) candidate 2-itemsets
    • To discover a frequent pattern of size 100, e.g., \(\{a_1, a_2, \ldots, a_{100}\}\), one needs to generate \(2^{100} \approx 10^{30}\) candidates.
  – Multiple scans of database:
    • Needs \((n+1)\) scans, \(n\) is the length of the longest pattern
Mining Frequent Patterns \textbf{Without Candidate Generation}

- Compress a large database into a compact, \textit{Frequent-}
  \textit{Pattern tree (FP-tree)} structure
  - highly condensed, but complete for frequent pattern mining
  - avoid costly database scans

- Develop an efficient, FP-tree-based frequent pattern mining method
  - A divide-and-conquer methodology: decompose mining tasks into smaller ones
  - Avoid candidate generation: sub-database test only
Construct FP-tree from a Transaction DB

<table>
<thead>
<tr>
<th>TID</th>
<th>Items bought</th>
<th>(ordered) frequent items</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>{f, a, c, d, g, i, m, p}</td>
<td>{f, c, a, m, p}</td>
</tr>
<tr>
<td>200</td>
<td>{a, b, c, f, l, m, o}</td>
<td>{f, c, a, b, m}</td>
</tr>
<tr>
<td>300</td>
<td>{b, f, h, j, o}</td>
<td>{f, b}</td>
</tr>
<tr>
<td>400</td>
<td>{b, c, k, s, p}</td>
<td>{c, b, p}</td>
</tr>
<tr>
<td>500</td>
<td>{a, f, c, e, l, p, m, n}</td>
<td>{f, c, a, m, p}</td>
</tr>
</tbody>
</table>

**min_support = 0.5**

Steps:

1. Scan DB once, find frequent 1-itemset (single item pattern)
2. Order frequent items in frequency descending order
3. Scan DB again, construct FP-tree
Benefits of the FP-tree Structure

• Completeness:
  – never breaks a long pattern of any transaction
  – preserves complete information for frequent pattern mining

• Compactness
  – reduce irrelevant information—in frequent items are gone
  – frequency descending ordering: more frequent items are more likely to be shared
  – never be larger than the original database
  – Example: For Connect-4 DB, compression ratio could be over 100
Mining Frequent Patterns Using FP-tree

- General idea (divide-and-conquer)
  - Recursively grow frequent pattern path using the FP-tree
- Method
  - For each item, construct its conditional pattern-base, and then its conditional FP-tree
  - Repeat the process on each newly created conditional FP-tree
  - Until the resulting FP-tree is empty, or it contains only one path
    (single path will generate all the combinations of its sub-paths, each of which is a frequent pattern)
Major Steps to Mine FP-tree

1) Construct conditional pattern base for each node in the FP-tree
2) Construct conditional FP-tree from each conditional pattern-base
3) Recursively mine conditional FP-trees and grow frequent patterns obtained so far
   • If the conditional FP-tree contains a single path, simply enumerate all the patterns
Step 1: From FP-tree to Conditional Pattern Base

- Starting at the frequent header table in the FP-tree
- Traverse the FP-tree by following the link of each frequent item
- Accumulate all of transformed prefix paths of that item to form a conditional pattern base
Properties of FP-tree for Conditional Pattern Base Construction

- **Node-link property**
  - For any frequent item $a_i$, all the possible frequent patterns that contain $a_i$ can be obtained by following $a_i$'s node-links, starting from $a_i$'s head in the FP-tree header.

- **Prefix path property**
  - To calculate the frequent patterns for a node $a_i$ in a path $P$, only the prefix sub-path of $a_i$ in $P$ need to be accumulated, and its frequency count should carry the same count as node $a_i$. 
Step 2: Construct Conditional FP-tree

- For each pattern-base
  - Accumulate the count for each item in the base
  - Construct the FP-tree for the frequent items of the pattern base
Mining Frequent Patterns by Creating Conditional Pattern-Bases

<table>
<thead>
<tr>
<th>Item</th>
<th>Conditional pattern-base</th>
<th>Conditional FP-tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>{((fca:2), (cb:1))}</td>
<td>{c:3}</td>
</tr>
<tr>
<td>m</td>
<td>{((fca:2), (fcab:1))}</td>
<td>{(f:3, c:3, a:3)}</td>
</tr>
<tr>
<td>b</td>
<td>{((fca:1), (f:1), (c:1))}</td>
<td>Empty</td>
</tr>
<tr>
<td>a</td>
<td>{((fc:3)}</td>
<td>{(f:3, c:3)}</td>
</tr>
<tr>
<td>c</td>
<td>{((f:3)}</td>
<td>{(f:3)}</td>
</tr>
<tr>
<td>f</td>
<td>Empty</td>
<td>Empty</td>
</tr>
</tbody>
</table>
Step 3: Recursively mine the conditional FP-tree

Cond. pattern base of “am”: (fc:3)

Cond. pattern base of “cm”: (f:3)

Cond. pattern base of “cam”: (f:3)
Single FP-tree Path Generation

- Suppose an FP-tree $T$ has a single path $P$
- The complete set of frequent pattern of $T$ can be generated by enumeration of all the combinations of the sub-paths of $P$

```
{}                      All frequent patterns concerning $m$
 f:3                    $m,$
c:3                     $fm, cm, am,$
a:3                     $fcm, fam, cam,$

m-conditional FP-tree
```
Principles of Frequent Pattern Growth

• Pattern growth property
  – Let $\alpha$ be a frequent itemset in DB, B be $\alpha$'s conditional pattern base, and $\beta$ be an itemset in B. Then $\alpha \cup \beta$ is a frequent itemset in DB iff $\beta$ is frequent in B.

• “abcdef” is a frequent pattern, if and only if
  – “abcde” is a frequent pattern, and
  – “f” is frequent in the set of transactions containing “abcde”
Why Is Frequent Pattern Growth Fast?

• Performance study shows
  – in some cases FP-growth is an order of magnitude faster than Apriori, and is also faster than tree-projection

• Reasoning
  – No candidate generation, no candidate test
  – Use compact data structure
  – Eliminate repeated database scan
  – Basic operation is counting and FP-tree building
FP-growth vs. Apriori: Scalability With the Support Threshold

Data set T25I20D10K
### Presentation of Association Rules

**Table Form**

<table>
<thead>
<tr>
<th>Body</th>
<th>Implies</th>
<th>Head</th>
<th>Supp (%)</th>
<th>Conf (%)</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>cost(x) = 0.00~1000.00</td>
<td>==&gt;</td>
<td>revenue(x) = 0.00~500.00</td>
<td>28.45</td>
<td>43.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cost(x) = 0.00~1000.00</td>
<td>==&gt;</td>
<td>revenue(x) = 500.00~1000.00</td>
<td>20.46</td>
<td>29.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cost(x) = 0.00~1000.00</td>
<td>==&gt;</td>
<td>order_qty(x) = 0.00~100.00</td>
<td>59.17</td>
<td>84.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cost(x) = 0.00~1000.00</td>
<td>==&gt;</td>
<td>revenue(x) = 1000.00~1500.00</td>
<td>10.45</td>
<td>14.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cost(x) = 0.00~1000.00</td>
<td>==&gt;</td>
<td>region(x) = United States</td>
<td>22.56</td>
<td>32.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cost(x) = 1000.00~2000.00</td>
<td>==&gt;</td>
<td>order_qty(x) = 0.00~100.00</td>
<td>12.91</td>
<td>69.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>order_qty(x) = 0.00~100.00</td>
<td>==&gt;</td>
<td>revenue(x) = 0.00~500.00</td>
<td>28.45</td>
<td>34.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>order_qty(x) = 0.00~100.00</td>
<td>==&gt;</td>
<td>cost(x) = 1000.00~2000.00</td>
<td>12.91</td>
<td>15.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>region(x) = United States</td>
<td>==&gt;</td>
<td>order_qty(x) = 0.00~100.00</td>
<td>25.9</td>
<td>31.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>order_qty(x) = 0.00~100.00</td>
<td>==&gt;</td>
<td>region(x) = United States</td>
<td>59.17</td>
<td>71.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>product_line(x) = &quot;Tents&quot;</td>
<td>==&gt;</td>
<td>order_qty(x) = 0.00~100.00</td>
<td>13.52</td>
<td>16.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>region(x) = United States</td>
<td>==&gt;</td>
<td>order_qty(x) = 0.00~100.00</td>
<td>25.9</td>
<td>81.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>revenue(x) = 0.00~500.00</td>
<td>==&gt;</td>
<td>cost(x) = 1000.00~1500.00</td>
<td>22.56</td>
<td>71.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>revenue(x) = 0.00~500.00</td>
<td>==&gt;</td>
<td>cost(x) = 0.00~100.00</td>
<td>28.45</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>revenue(x) = 1000.00~1500.00</td>
<td>==&gt;</td>
<td>order_qty(x) = 0.00~100.00</td>
<td>28.45</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>revenue(x) = 500.00~1000.00</td>
<td>==&gt;</td>
<td>cost(x) = 0.00~1000.00</td>
<td>10.45</td>
<td>96.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>revenue(x) = 500.00~1000.00</td>
<td>==&gt;</td>
<td>order_qty(x) = 0.00~100.00</td>
<td>20.46</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>revenue(x) = 0.00~1000.00</td>
<td>==&gt;</td>
<td>revenue(x) = 0.00<del>500.00 AND order_qty(x) = 0.00</del>100.00</td>
<td>28.45</td>
<td>40.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>revenue(x) = 0.00~1000.00</td>
<td>==&gt;</td>
<td>revenue(x) = 0.00<del>500.00 AND order_qty(x) = 0.00</del>100.00</td>
<td>28.45</td>
<td>40.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>revenue(x) = 0.00~1000.00</td>
<td>==&gt;</td>
<td>revenue(x) = 500.00<del>1000.00 AND order_qty(x) = 0.00</del>100.00</td>
<td>19.67</td>
<td>27.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>revenue(x) = 0.00~1000.00</td>
<td>==&gt;</td>
<td>revenue(x) = 500.00<del>1000.00 AND order_qty(x) = 0.00</del>100.00</td>
<td>19.67</td>
<td>27.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Visualization of Association Rule Using Plane Graph

Promotions = [No Promotion] \implies \text{Gender = [M], support: 37.06\%, confidence: 50.55\%]
Visualization of Association Rule Using Rule Graph
Iceberg Queries

- **Iceberg query**: Compute aggregates over one or a set of attributes only for those whose aggregate values is above a certain threshold.

- Example:
  ```sql
  select P.custID, P.itemID, sum(P.qty)
  from purchase P
  group by P.custID, P.itemID
  having sum(P.qty) >= 10
  ```

- Compute iceberg queries efficiently by **Apriori**:
  - First compute lower dimensions
  - Then compute higher dimensions only when all the lower ones are above the threshold
Multiple-Level Association Rules

- Items often form hierarchy.
- Items at the lower level are expected to have lower support.
- Rules regarding itemsets at appropriate levels could be quite useful.
- Transaction database can be encoded based on dimensions and levels
- We can explore shared multi-level mining

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>{111, 121, 211, 221}</td>
</tr>
<tr>
<td>T2</td>
<td>{111, 211, 222, 323}</td>
</tr>
<tr>
<td>T3</td>
<td>{112, 122, 221, 411}</td>
</tr>
<tr>
<td>T4</td>
<td>{111, 121}</td>
</tr>
<tr>
<td>T5</td>
<td>{111, 122, 211, 221, 413}</td>
</tr>
</tbody>
</table>
Mining Multi-Level Associations

• A top_down, progressive deepening approach:
  – First find high-level strong rules:
    milk → bread [20%, 60%].
  – Then find their lower-level “weaker” rules:
    2% milk → wheat bread [6%, 50%].

• Variations at mining multiple-level association rules.
  – Level-crossed association rules:
    2% milk → Wonder wheat bread
  – Association rules with multiple, alternative hierarchies:
    2% milk → Wonder bread
Multi-level Association: Uniform Support vs. Reduced Support

• Uniform Support: the same minimum support for all levels
  – + One minimum support threshold. No need to examine itemsets containing any item whose ancestors do not have minimum support.
  – – Lower level items do not occur as frequently. If support threshold
    • too high \( \Rightarrow \) miss low level associations
    • too low \( \Rightarrow \) generate too many high level associations

• Reduced Support: reduced minimum support at lower levels
  – There are 4 search strategies:
    • Level-by-level independent
    • Level-cross filtering by k-itemset
    • Level-cross filtering by single item
    • Controlled level-cross filtering by single item
Uniform Support

Multi-level mining with uniform support

Level 1
\text{min}\_\text{sup} = 5\% 

Level 2
\text{min}\_\text{sup} = 5\% 

Milk
\text{[support} = 10\%]\text{]}

2\% Milk
\text{[support} = 6\%]\text{]}

Skim Milk
\text{[support} = 4\%]\text{]}

Back
Reduced Support

Multi-level mining with reduced support

Level 1
min_sup = 5%

Milk
[support = 10%]

Level 2
min_sup = 3%

2% Milk
[support = 6%]

Skim Milk
[support = 4%]
Multi-level Association: Redundancy Filtering

• Some rules may be redundant due to “ancestor” relationships between items.

• Example
  – milk ⇒ wheat bread  [support = 8%, confidence = 70%]
  – 2% milk ⇒ wheat bread [support = 2%, confidence = 72%]

• We say the first rule is an ancestor of the second rule.

• A rule is redundant if its support is close to the “expected” value, based on the rule’s ancestor.
Multi-Level Mining: Progressive Deepening

- A top-down, progressive deepening approach:
  - First mine high-level frequent items: milk (15%), bread (10%)
  - Then mine their lower-level “weaker” frequent itemsets: 2% 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.
Progressive Refinement of Data Mining Quality

• Why progressive refinement?
  – 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).
Multi-Dimensional Association: Concepts

- **Single-dimensional rules:**
  \[ \text{buys}(X, "milk") \Rightarrow \text{buys}(X, "bread") \]

- **Multi-dimensional rules:** \( \bigcirc \) 2 dimensions or predicates
  - Inter-dimension association rules (*no repeated predicates*)
    \[ \text{age}(X, "19-25") \land \text{occupation}(X, "student") \Rightarrow \text{buys}(X, "coke") \]
  - Hybrid-dimension association rules (*repeated predicates*)
    \[ \text{age}(X, "19-25") \land \text{buys}(X, "popcorn") \Rightarrow \text{buys}(X, "coke") \]

- **Categorical Attributes**
  - finite number of possible values, no ordering among values

- **Quantitative Attributes**
  - numeric, implicit ordering among values
Techniques for Mining MD Associations

- Search for frequent $k$-predicate set:
  - Example: \{age, occupation, buys\} is a 3-predicate set.
  - Techniques can be categorized by how age are treated.

1. Using static discretization of quantitative attributes
   - Quantitative attributes are statically discretized by using predefined concept hierarchies.

2. Quantitative association rules
   - Quantitative attributes are dynamically discretized into “bins” based on the distribution of the data.

3. Distance-based association rules
   - This is a dynamic discretization process that considers the distance between data points.
Static Discretization of Quantitative Attributes

- Discretized prior to mining using concept hierarchy.
- Numeric values are replaced by ranges.
- In relational database, finding all frequent k-predicate sets will require $k$ or $k+1$ table scans.
- Data cube is well suited for mining.
- The cells of an n-dimensional cuboid correspond to the predicate sets.
- Mining from data cubes can be much faster.
Quantitative Association Rules

- Numeric attributes are *dynamically* discretized
  - Such that the confidence or compactness of the rules mined is maximized.
- 2-D quantitative association rules: $A_{\text{quan1}} \land A_{\text{quan2}} \Rightarrow A_{\text{cat}}$
- Cluster “adjacent” association rules to form general rules using a 2-D grid.
- Example:

\[
\text{age}(X, "30-34") \land \text{income}(X, "24K - 48K") \\
\Rightarrow \text{buys}(X, "high resolution TV")
\]
Clusters and Distance Measurements

- $S[X]$ is a set of $N$ tuples $t_1, t_2, \ldots, t_N$, projected on the attribute set $X$.
- The diameter of $S[X]$: 

$$d(S[X]) = \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \text{dist}_x(t_i[X], t_j[X])}{N(N-1)}$$

- $\text{dist}_x$: distance metric, e.g. Euclidean distance or Manhattan.
Clusters and Distance Measurements (Cont.)

• The diameter, $d$, assesses the density of a cluster $C_x$, where

$$d(C_x) \leq d_0^x$$

$$|C_x| \geq s_0$$

• Finding clusters and distance-based rules
  – the density threshold, $d_0$, replaces the notion of support
  – modified version of the BIRCH clustering algorithm
Interestingness Measurements

- **Objective measures**
  - Two popular measurements:
    - \( \star \) support; and
    - \( \circ \) confidence

- **Subjective measures** (Silberschatz & Tuzhilin, KDD95)
  - A rule (pattern) is interesting if
  - \( \star \) it is *unexpected* (surprising to the user); and/or
  - \( \circ \) *actionable* (the user can do something with it)
Criticism to Support and Confidence

- Example 1: (Aggarwal & Yu, PODS98)
  - Among 5000 students
    - 3000 play basketball
    - 3750 eat cereal
    - 2000 both play basket ball and eat cereal
  - \(\text{play basketball} \Rightarrow \text{eat cereal} \) [40%, 66.7%] is misleading because the overall percentage of students eating cereal is 75% which is higher than 66.7%.
  - \(\text{play basketball} \Rightarrow \text{not eat cereal} \) [20%, 33.3%] is far more accurate, although with lower support and confidence

<table>
<thead>
<tr>
<th></th>
<th>basketball</th>
<th>not basketball</th>
<th>sum(row)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cereal</td>
<td>2000</td>
<td>1750</td>
<td>3750</td>
</tr>
<tr>
<td>not cereal</td>
<td>1000</td>
<td>250</td>
<td>1250</td>
</tr>
<tr>
<td>sum(col.)</td>
<td>3000</td>
<td>2000</td>
<td>5000</td>
</tr>
</tbody>
</table>
• Example 2:
  – X and Y: positively correlated,
  – X and Z, negatively related
  – support and confidence of X=>Z dominates

• We need a measure of dependent or correlated events

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

• \(P(B|A)/P(B)\) is also called the lift of rule \(A \Rightarrow B\)
Other Interestingness Measures: Interest

- Interest (correlation, lift)
  \[
  \frac{P(A \land B)}{P(A)P(B)}
  \]
  - taking both \( P(A) \) and \( P(B) \) in consideration
  - \( P(A \land B) = P(B)*P(A) \), if \( A \) and \( B \) are independent events
  - \( A \) and \( B \) negatively correlated, if the value is less than 1; otherwise \( A \) and \( B \) positively correlated

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Support</th>
<th>Interest</th>
</tr>
</thead>
<tbody>
<tr>
<td>X,Y</td>
<td>25%</td>
<td>2</td>
</tr>
<tr>
<td>X,Z</td>
<td>37.50%</td>
<td>0.9</td>
</tr>
<tr>
<td>Y,Z</td>
<td>12.50%</td>
<td>0.57</td>
</tr>
</tbody>
</table>
Viewing
Association Rules
## Presentation of Association Rules

### (Table Form)

<table>
<thead>
<tr>
<th></th>
<th>Body</th>
<th>Implies</th>
<th>Head</th>
<th>Supp (%)</th>
<th>Conf (%)</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>cost(x) = 0.00-1000.00</td>
<td>=&gt; revenue(x) = 0.00-500.00</td>
<td></td>
<td>28.45</td>
<td>40.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>cost(x) = 0.00-1000.00</td>
<td>=&gt; revenue(x) = '500.00-1000.00'</td>
<td></td>
<td>20.46</td>
<td>29.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>cost(x) = 0.00-1000.00</td>
<td>=&gt; order_qty(x) = 0.00-100.00</td>
<td></td>
<td>59.17</td>
<td>84.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>cost(x) = 0.00-1000.00</td>
<td>=&gt; revenue(x) = '1000.00-1500.00'</td>
<td></td>
<td>10.45</td>
<td>14.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>cost(x) = 0.00-1000.00</td>
<td>=&gt; region(x) = 'United States'</td>
<td></td>
<td>22.56</td>
<td>32.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>cost(x) = '1000.00-2000.00'</td>
<td>=&gt; order_qty(x) = 0.00-100.00</td>
<td></td>
<td>12.91</td>
<td>69.34</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>order_qty(x) = 0.00-100.00</td>
<td>=&gt; revenue(x) = 0.00-500.00</td>
<td></td>
<td>28.45</td>
<td>34.54</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>order_qty(x) = 0.00-100.00</td>
<td>=&gt; cost(x) = '1000.00-2000.00'</td>
<td></td>
<td>12.91</td>
<td>15.67</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>order_qty(x) = 0.00-100.00</td>
<td>=&gt; region(x) = 'United States'</td>
<td></td>
<td>25.9</td>
<td>31.45</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>order_qty(x) = 0.00-100.00</td>
<td>=&gt; cost(x) = '500.00-1000.00'</td>
<td></td>
<td>59.17</td>
<td>71.86</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>order_qty(x) = 0.00-100.00</td>
<td>=&gt; product_line(x) = 'Tents'</td>
<td></td>
<td>13.52</td>
<td>16.42</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>order_qty(x) = 0.00-100.00</td>
<td>=&gt; revenue(x) = '500.00-1000.00'</td>
<td></td>
<td>19.67</td>
<td>23.88</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>product_line(x) = 'Tents'</td>
<td>=&gt; order_qty(x) = 0.00-100.00</td>
<td></td>
<td>13.52</td>
<td>93.72</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>region(x) = 'United States'</td>
<td>=&gt; order_qty(x) = 0.00-100.00</td>
<td></td>
<td>25.9</td>
<td>81.94</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>region(x) = 'United States'</td>
<td>=&gt; cost(x) = '0.00-100.00'</td>
<td></td>
<td>22.56</td>
<td>71.39</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>revenue(x) = '500.00-5000.00'</td>
<td>=&gt; cost(x) = '0.00-1000.00'</td>
<td></td>
<td>28.45</td>
<td>100.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>revenue(x) = '0.00-500.00'</td>
<td>=&gt; order_qty(x) = 0.00-100.00</td>
<td></td>
<td>28.45</td>
<td>100.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>revenue(x) = '1000.00-1500.00'</td>
<td>=&gt; cost(x) = '0.00-1000.00'</td>
<td></td>
<td>28.45</td>
<td>100.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>revenue(x) = '500.00-1000.00'</td>
<td>=&gt; cost(x) = '0.00-1000.00'</td>
<td></td>
<td>20.46</td>
<td>100.00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>revenue(x) = '500.00-1000.00'</td>
<td>=&gt; order_qty(x) = 0.00-100.00</td>
<td></td>
<td>19.67</td>
<td>96.14</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>cost(x) = 0.00-1000.00</td>
<td>=&gt; revenue(x) = 0.00-500.00 AND order_qty(x) = 0.00-100.00</td>
<td></td>
<td>28.45</td>
<td>40.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>cost(x) = 0.00-1000.00</td>
<td>=&gt; revenue(x) = 0.00-500.00 AND order_qty(x) = 0.00-100.00</td>
<td></td>
<td>28.45</td>
<td>40.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>cost(x) = 0.00-1000.00</td>
<td>=&gt; revenue(x) = '500.00-1000.00' AND order_qty(x) = 0.00-100.00</td>
<td></td>
<td>19.67</td>
<td>27.93</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>cost(x) = 0.00-1000.00</td>
<td>=&gt; revenue(x) = '500.00-1000.00' AND order_qty(x) = 0.00-100.00</td>
<td></td>
<td>19.67</td>
<td>27.93</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>cost(x) = 0.00-1000.00 AND order_qty(x) = 0.00-100.00</td>
<td>=&gt; revenue(x) = '500.00-1000.00'</td>
<td></td>
<td>19.67</td>
<td>33.23</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Visualization of Association Rule Using Plane Graph

Promotions = [No Promotion] ==> Gender = [M] | support: 37.06% | confidence: 50.55%
Visualization of Association Rule Using Rule Graph
References


• Notes from Jeff Ullman’s Data mining course, [http://www-db.stanford.edu/~ullman/mining/mining.html](http://www-db.stanford.edu/~ullman/mining/mining.html)

• Vipin Kumar and Mahesh Joshi’s tutorial on High Performance Data Mining, [http://www-users.cs.umn.edu/~mjoshi/hpdmtut/](http://www-users.cs.umn.edu/~mjoshi/hpdmtut/)