Association Analysis
Part 1

Dr. Sanjay Ranka
Professor
Computer and Information Science and Engineering
University of Florida
Mining Associations

• Given a set of records, find rules that will predict the occurrence of an item based on the occurrences of other items in the record.

Market-Basket transactions

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bread, Milk</td>
</tr>
<tr>
<td>2</td>
<td>Bread, Diaper, Beer, Eggs</td>
</tr>
<tr>
<td>3</td>
<td>Milk, Diaper, Beer, Coke</td>
</tr>
<tr>
<td>4</td>
<td>Bread, Milk, Diaper, Beer</td>
</tr>
<tr>
<td>5</td>
<td>Bread, Milk, Diaper, Coke</td>
</tr>
</tbody>
</table>

Example:

<table>
<thead>
<tr>
<th>TID</th>
<th>Bread</th>
<th>Milk</th>
<th>Diaper</th>
<th>Beer</th>
<th>Eggs</th>
<th>Coke</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
Definition of Association Rule

**Association Rule:** $X \Rightarrow y$

**Support:** $s = \frac{\sigma(X \cup y)}{|T|} (s = P(X, y))$

**Confidence:** $c = \frac{\sigma(X \cup y)}{\sigma(X)} (c = P(y | X))$

**Example:** $\{\text{Milk, Diaper}\} \Rightarrow \text{Beer}$

- $s = \frac{\sigma(\text{Milk, Diaper, Beer})}{|T|} = \frac{2}{5} = 0.4$
- $c = \frac{\sigma(\text{Milk, Diaper, Beer})}{\sigma(\text{Milk, Diaper})} = \frac{2}{3} = 0.67$

**Goal:**
Discover all rules having support $\geq \text{minsup}$ and confidence $\geq \text{minconf}$ thresholds.
How to Mine Association Rules?

Example of Rules:

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bread, Milk</td>
</tr>
<tr>
<td>2</td>
<td>Bread, Diaper, Beer, Eggs</td>
</tr>
<tr>
<td>3</td>
<td>Milk, Diaper, Beer, Coke</td>
</tr>
<tr>
<td>4</td>
<td>Bread, Milk, Diaper, Beer</td>
</tr>
<tr>
<td>5</td>
<td>Bread, Milk, Diaper, Coke</td>
</tr>
</tbody>
</table>

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

Observations:

• All the rules above correspond to the same itemset: {Milk, Diaper, Beer}

• Rules obtained from the same itemset have identical support but can have different confidence
How to Mine Association Rules?

• Two step approach:
  1. Generate all frequent itemsets (sets of items whose support > \textit{minsup})
  2. Generate high confidence association rules from each frequent itemset
     - Each rule is a binary partition of a frequent itemset

- Frequent itemset generation is more expensive operation
There are $2^d$ possible itemsets
Generating Frequent Itemsets

• Naive approach:
  – Each itemset in the lattice is a candidate frequent itemset
  – Count the support of each candidate by scanning the database

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bread, Milk</td>
</tr>
<tr>
<td>2</td>
<td>Bread, Diaper, Beer, Eggs</td>
</tr>
<tr>
<td>3</td>
<td>Milk, Diaper, Beer, Coke</td>
</tr>
<tr>
<td>4</td>
<td>Bread, Milk, Diaper, Beer</td>
</tr>
<tr>
<td>5</td>
<td>Bread, Milk, Diaper, Coke</td>
</tr>
</tbody>
</table>

- Complexity $\sim O(NM) \Rightarrow$ Expensive since $M = 2^d$ !!!

Data Mining, Sanjay Ranka  Spring 2011
Computational Complexity

• Given d unique items:
  - Total number of itemsets = \(2^d\)
  - Total number of possible association rules:

\[
R = \sum_{k=1}^{d-1} \left[ \binom{d}{k} \times \sum_{j=1}^{d-k} \binom{d-k}{j} \right] = 3^d - 2^{d+1} + 1
\]

If d=6, \(R = 602\) rules
Approach for Mining Frequent Itemsets

• Reduce the number of candidates \((M)\)
  – Complete search: \(M=2^d\)
  – Use Apriori heuristic to reduce \(M\)

• Reduce the number of transactions \((N)\)
  – Reduce size of \(N\) as the size of itemset increases
  – Used by DHP and vertical-based mining algorithms

• Reduce the number of comparisons \((NM)\)
  – Use efficient data structures to store the candidates or transactions
  – No need to match every candidate against every transaction
Reducing Number of Candidates

• Apriori principle:
  – If an itemset is frequent, then all of its subsets must also be frequent

• Apriori principle holds due to the following property of the support measure:
  \[ \forall X, Y : (X \subseteq Y) \Rightarrow \sigma(X) \geq \sigma(Y) \]
  – Support of an itemset never exceeds the support of any of its subsets
  – This is known as the **anti-monotone** property of support
Using Apriori Principle for Pruning Candidates

If an itemset is infrequent, then all of its supersets must also be infrequent

Data Mining, Sanjay Ranka  Spring 2011
Illustrating Apriori Principle

<table>
<thead>
<tr>
<th>Item</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bread</td>
<td>4</td>
</tr>
<tr>
<td>Coke</td>
<td>2</td>
</tr>
<tr>
<td>Milk</td>
<td>4</td>
</tr>
<tr>
<td>Beer</td>
<td>3</td>
</tr>
<tr>
<td>Diaper</td>
<td>4</td>
</tr>
<tr>
<td>Eggs</td>
<td>1</td>
</tr>
</tbody>
</table>

Minimum Support = 3

If every subset is considered,
$$6C_1 + 6C_2 + 6C_3 = 41$$

With support-based pruning,
$$6 + 6 + 1 = 13$$

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Bread,Milk}</td>
<td>3</td>
</tr>
<tr>
<td>{Bread,Beer}</td>
<td>2</td>
</tr>
<tr>
<td>{Bread,Diaper}</td>
<td>3</td>
</tr>
<tr>
<td>{Milk,Beer}</td>
<td>2</td>
</tr>
<tr>
<td>{Milk,Diaper}</td>
<td>3</td>
</tr>
<tr>
<td>{Beer,Diaper}</td>
<td>3</td>
</tr>
</tbody>
</table>

Pairs (2-itemsets)

(No need to generate candidates involving Coke or Eggs)

<table>
<thead>
<tr>
<th>Itemset</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>{Bread,Milk,Diaper}</td>
<td>3</td>
</tr>
</tbody>
</table>

Triplets (3-itemsets)
Reducing Number of Comparisons

• Candidate counting:
  – Scan the database of transactions to determine the support of candidate itemsets
  – To reduce number of comparisons, store the candidates using a hash structure

<table>
<thead>
<tr>
<th>TID</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bread, Milk</td>
</tr>
<tr>
<td>2</td>
<td>Bread, Diaper, Beer, Eggs</td>
</tr>
<tr>
<td>3</td>
<td>Milk, Diaper, Beer, Coke</td>
</tr>
<tr>
<td>4</td>
<td>Bread, Milk, Diaper, Beer</td>
</tr>
<tr>
<td>5</td>
<td>Bread, Milk, Diaper, Coke</td>
</tr>
</tbody>
</table>
Association Rule Discovery: Hash Tree for Fast Access

Hash Function

Candidate Hash Tree

Hash on 1, 4 or 7

Data Mining, Sanjay Ranka, Spring 2011
Association Rule Discovery: Hash Tree for Fast Access

Hash Function

1, 4, 7
2, 5, 8
3, 6, 9

Candidate Hash Tree

Hash on 2, 5 or 8

1, 2, 4, 5, 7
2, 5, 8
3, 6, 9
Association Rule Discovery: Hash Tree for Fast Access

Hash Function

1, 4, 7
2, 5, 8
3, 6, 9

Candidate Hash Tree

Hash on 3, 6 or 9

1, 2, 4
1, 2, 5
1, 4, 7
2, 5, 8
2, 5, 9
3, 5, 6
3, 5, 7
3, 5, 8
3, 6, 7
3, 6, 8
Candidate Counting

- Given a transaction \( L = \{1,2,3,5,6\} \)
- Possible subsets of size 3:
  - \( \{1,2,3\} \)
  - \( \{2,3,5\} \)
  - \( \{3,5,6\} \)
  - \( \{1,2,5\} \)
  - \( \{2,3,6\} \)
  - \( \{1,2,6\} \)
  - \( \{2,5,6\} \)
  - \( \{1,3,5\} \)
  - \( \{2,5,6\} \)
  - \( \{1,3,6\} \)
  - \( \{1,5,6\} \)

- If width of transaction is \( w \), there are \( 2^w - 1 \) possible non-empty subsets
Association Rule Discovery: Subset Operation

![Diagram of association rule discovery with subset operation]

- **Transaction:** 1 2 3 5 6
- **Hash Function:**
  - 1,4,7
  - 2,5,8
  - 3,6,9

- **Subset Operations:**
  - 1 + 2 3 5 6
  - 2 + 3 5 6
  - 3 + 5 6

- **Transactions:**
  - 1 2 4
  - 4 5 7
  - 1 2 5
  - 4 5 8
  - 1 2 3 5 6
  - 1 + 2 3 5 6
  - 2 + 3 5 6
  - 3 + 5 6

- **Sets:**
  - 1 4 5
  - 1 3 6
  - 2 3 4
  - 5 6 7
  - 1 2 4
  - 4 5 7
  - 1 2 5
  - 4 5 8
  - 1 2 3 5 6
  - 1 + 2 3 5 6
  - 2 + 3 5 6
  - 3 + 5 6
  - 3 4 5
  - 3 5 6
  - 3 5 7
  - 3 6 7
  - 3 6 8
  - 3 5 6
  - 3 5 7
  - 3 6 9
  - 3 6 8

Data Mining, Sanjay Ranka, Spring 2011
Association Rule Discovery: Subset Operation …
Rule Generation

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

  – If \{A,B,C,D\} is a frequent itemset, candidate rules:

    \[
    \begin{align*}
    ABC & \rightarrow D, & ABD & \rightarrow C, & ACD & \rightarrow B, & BCD & \rightarrow A, \\
    A & \rightarrow BCD, & B & \rightarrow ACD, & C & \rightarrow ABD, & D & \rightarrow ABC, \\
    AB & \rightarrow CD, & AC & \rightarrow BD, & AD & \rightarrow BC, & BC & \rightarrow AD, \\
    BD & \rightarrow AC, & CD & \rightarrow AB, & & & &
    \end{align*}
    \]

• If \(|L| = k\), then there are \(2^k - 2\) candidate association rules (ignoring \( L \rightarrow \emptyset \) and \( \emptyset \rightarrow L \))
Rule Generation

• How to efficiently generate rules from frequent itemsets?
  – In general, confidence does not have an anti-monotone property
  – But confidence of rules generated from the same itemset has an anti-monotone property
  – \( L = \{A, B, C, D\}:\)
    \[
    c(ABC \rightarrow D) \geq c(AB \rightarrow CD) \geq c(A \rightarrow BCD)
    \]

• Confidence is non-increasing as number of items in rule consequent increases
Rule Generation for Apriori Algorithm

Lattice of rules

- Lattice corresponds to partial order of items in the rule consequent
Rule Generation for Apriori Algorithm …

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

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

• Prune rule $D => ABC$ if its subset $AD => BC$ does not have high confidence
Other Frequent Itemset Algorithms

• Traversal of Itemset Lattice
  – Apriori uses breadth-first (level-wise) traversal

• Representation of Database
  – Apriori uses horizontal data layout

• Generate-and-count paradigm