Association Rules Outline

**Goal:** Provide an overview of basic Association Rule mining techniques

- Association Rules Problem Overview
  - Large/Frequent itemsets
- Association Rules Algorithms
  - Apriori
  - Sampling
  - Partitioning
  - Parallel Algorithms
- Comparing Techniques
- Incremental Algorithms
- Advanced AR Techniques
Example: Market Basket Data

• Items frequently purchased together:
  \textbf{Bread }\Rightarrow\textbf{PeanutButter}

• Uses:
  – Product placement
  – Advertising - Amazon
  – Sales
  – Coupons
Association Rule Definitions

• **Set of items:** $I = \{I_1, I_2, \ldots, I_m\}$
• **Transactions:** $D = \{t_1, t_2, \ldots, t_n\}, t_j \subseteq I$
• **Itemset:** $\{I_{i1}, I_{i2}, \ldots, I_{ik}\} \subseteq I$
• **Support of an itemset:** Percentage of transactions which contain that itemset.
• **Large (Frequent) itemset:** Itemset whose number of occurrences is above a threshold.
Association Rules Example

I = \{ Beer, Bread, Jelly, Milk, PeanutButter\}

Support of \{Bread, PeanutButter\} is 60%
 Association Rule Definitions

- **Association Rule (AR):** implication \( X \Rightarrow Y \) where \( X, Y \subseteq I \) and \( X \cap Y = \emptyset \);

- **Support of AR (s) \( X \Rightarrow Y \):** Percentage of transactions that contain \( X \cup Y \)

- **Confidence of AR (\( \alpha \) \( X \Rightarrow Y \):** Ratio of number of transactions that contain \( X \cup Y \) to the number that contain \( X \).
Association Rules Ex (cont’d)

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>Bread, Jelly, PeanutButter</td>
</tr>
<tr>
<td>$t_2$</td>
<td>Bread, PeanutButter</td>
</tr>
<tr>
<td>$t_3$</td>
<td>Bread, Milk, PeanutButter</td>
</tr>
<tr>
<td>$t_4$</td>
<td>Beer, Bread</td>
</tr>
<tr>
<td>$t_5$</td>
<td>Beer, Milk</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$X \Rightarrow Y$</th>
<th>$s$</th>
<th>$\alpha$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bread $\Rightarrow$ PeanutButter</td>
<td>60%</td>
<td>75%</td>
</tr>
<tr>
<td>PeanutButter $\Rightarrow$ Bread</td>
<td>60%</td>
<td>100%</td>
</tr>
<tr>
<td>Beer $\Rightarrow$ Bread</td>
<td>20%</td>
<td>50%</td>
</tr>
<tr>
<td>PeanutButter $\Rightarrow$ Jelly</td>
<td>20%</td>
<td>33.3%</td>
</tr>
<tr>
<td>Jelly $\Rightarrow$ PeanutButter</td>
<td>20%</td>
<td>100%</td>
</tr>
<tr>
<td>Jelly $\Rightarrow$ Milk</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>
Association Rule Problem

• Given a set of items $I=\{I_1,I_2,\ldots,I_m\}$ and a database of transactions $D=\{t_1,t_2, \ldots, t_n\}$ where $t_i=\{I_{i1},I_{i2}, \ldots, I_{ik}\}$ and $I_{ij} \in I$, the **Association Rule Problem** is to identify all association rules $X \Rightarrow Y$ with a minimum support and confidence.

• Link Analysis

• **NOTE**: Support of $X \Rightarrow Y$ is same as support of $X \cup Y$. 
Association Rule Techniques

1. Find Large Itemsets.
2. Generate rules from frequent itemsets.
Algorithm to Generate ARs

Input:
- $D$ //Database of transactions
- $I$ //Items
- $L$ //Large itemsets
- $s$ //Support
- $\alpha$ //Confidence

Output:
- $R$ //Association Rules satisfying $s$ and $\alpha$

ARGen Algorithm:
1. $R = \emptyset$;
2. for each $l \in L$ do
   - for each $x \subset l$ such that $x \neq \emptyset$ and $x \neq l$ do
     - if $\frac{\text{support}(l)}{\text{support}(x)} \geq \alpha$ then
       - $R = R \cup \{x \Rightarrow (l - x)\}$;
Apriori

**Large Itemset Property:**
Any subset of a large itemset is large.

**Contrapositive:**
If an itemset is not large, none of its supersets are large.
Large Itemset Property
### Apriori Ex (cont’d)

<table>
<thead>
<tr>
<th>Pass</th>
<th>Candidates</th>
<th>Large Itemsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{Beer}, {Bread}, {Jelly}, {Milk}, {PeanutButter}</td>
<td>{Beer}, {Bread}, {Milk}, {PeanutButter}</td>
</tr>
<tr>
<td>2</td>
<td>{Beer, Bread}, {Beer, Milk}, {Beer, PeanutButter}, {Bread, Milk}, {Bread, PeanutButter}, {Milk, PeanutButter}</td>
<td>{Bread, PeanutButter}</td>
</tr>
</tbody>
</table>

s = 30% \quad \alpha = 50%
Apriori Algorithm

1. \( C_1 = \) Itemsets of size one in \( I \);
2. Determine all large itemsets of size 1, \( L_1 \);
3. \( i = 1 \);
4. Repeat
5. \( i = i + 1 \);
6. \( C_i = \) Apriori-Gen\((L_{i-1})\);
7. Count \( C_i \) to determine \( L_i \);
8. until no more large itemsets found;
Apriori-Gen

• Generate candidates of size i+1 from large itemsets of size i.
• Approach used: join large itemsets of size i if they agree on i-1
• May also prune candidates who have subsets that are not large.
## Apriori-Gen Example

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>Blouse</td>
</tr>
<tr>
<td>$t_2$</td>
<td>Shoes,Skirt,TShirt</td>
</tr>
<tr>
<td>$t_3$</td>
<td>Jeans,TShirt</td>
</tr>
<tr>
<td>$t_4$</td>
<td>Jeans,Shoes,TShirt</td>
</tr>
<tr>
<td>$t_5$</td>
<td>Jeans,Shorts</td>
</tr>
<tr>
<td>$t_6$</td>
<td>Shoes,TShirt</td>
</tr>
<tr>
<td>$t_7$</td>
<td>Jeans,Skirt</td>
</tr>
<tr>
<td>$t_8$</td>
<td>Jeans,Shoes,Shorts,TShirt</td>
</tr>
<tr>
<td>$t_9$</td>
<td>Jeans</td>
</tr>
<tr>
<td>$t_{10}$</td>
<td>Jeans,Shoes,TShirt</td>
</tr>
<tr>
<td>$t_{11}$</td>
<td>TShirt</td>
</tr>
<tr>
<td>$t_{12}$</td>
<td>Blouse,Jeans,Shoes,Skirt,TShirt</td>
</tr>
<tr>
<td>$t_{13}$</td>
<td>Jeans,Shoes,Shorts,TShirt</td>
</tr>
<tr>
<td>$t_{14}$</td>
<td>Shoes,Skirt,TShirt</td>
</tr>
<tr>
<td>$t_{15}$</td>
<td>Jeans,TShirt</td>
</tr>
<tr>
<td>$t_{16}$</td>
<td>Skirt,TShirt</td>
</tr>
<tr>
<td>$t_{17}$</td>
<td>Blouse,Jeans,Skirt</td>
</tr>
<tr>
<td>$t_{18}$</td>
<td>Jeans,Shoes,Shorts,TShirt</td>
</tr>
<tr>
<td>$t_{19}$</td>
<td>Jeans</td>
</tr>
<tr>
<td>$t_{20}$</td>
<td>Jeans,Shoes,Shorts,TShirt</td>
</tr>
</tbody>
</table>
### Apriori-Gen Example (cont’d)

<table>
<thead>
<tr>
<th>Scan</th>
<th>Candidates</th>
<th>Large Itemsets</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{Blouse}, {Jeans}, {Shoes}, {Shorts}, {Skirt}, {TShirt}</td>
<td>{Jeans}, {Shoes}, {Shorts}, {Skirt}, {TShirt}</td>
</tr>
<tr>
<td>4</td>
<td>{Jeans, Shoes, Shorts, TShirt}</td>
<td>{Jeans, Shoes, Shorts, TShirt}</td>
</tr>
<tr>
<td>5</td>
<td>∅</td>
<td>∅</td>
</tr>
</tbody>
</table>
Apriori Adv/Disadv

- **Advantages:**
  - Uses large itemset property.
  - Easily parallelized. How?
  - Easy to implement.

- **Disadvantages:**
  - Assumes transaction database is memory resident.
  - Requires up to m database scans.
Partitioning

- Divide database into partitions $D^1, D^2, \ldots, D^p$
- Apply Apriori to each partition
- Any large itemset must be large in at least one partition.
Partitioning Algorithm

1. Divide $D$ into partitions $D^1, D^2, \ldots, D^p$;
2. For $I = 1$ to $p$ do
3. \[ L^i = \text{Apriori}(D^i); \]
4. \[ C = L^1 \cup \ldots \cup L^p; \]
5. Count $C$ on $D$ to generate $L$;
## Partitioning Example

<table>
<thead>
<tr>
<th>Transaction</th>
<th>Items</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t_1$</td>
<td>Bread, Jelly, PeanutButter</td>
</tr>
<tr>
<td>$t_2$</td>
<td>Bread, PeanutButter</td>
</tr>
<tr>
<td>$t_3$</td>
<td>Bread, Milk, PeanutButter</td>
</tr>
<tr>
<td>$t_4$</td>
<td>Beer, Bread</td>
</tr>
<tr>
<td>$t_5$</td>
<td>Beer, Milk</td>
</tr>
</tbody>
</table>

$L^1 = \{\{\text{Bread}\}, \{\text{Jelly}\}, \{\text{PeanutButter}\}, \{\text{Bread, Jelly}\}, \{\text{Bread, PeanutButter}\}, \{\text{Jelly, PeanutButter}\}, \{\text{Bread, Jelly, PeanutButter}\}\}$

$L^2 = \{\{\text{Bread}\}, \{\text{Milk}\}, \{\text{PeanutButter}\}, \{\text{Bread, Milk}\}, \{\text{Bread, PeanutButter}\}, \{\text{Milk, PeanutButter}\}, \{\text{Bread, Milk, PeanutButter}\}, \{\text{Beer}\}, \{\text{Beer, Bread}\}, \{\text{Beer, Milk}\}\}$
Partitioning Adv/Disadv

• **Advantages:**
  – Adapts to available main memory
  – Easily parallelized
  – Maximum number of database scans is two.

• **Disadvantages:**
  – May have many candidates during second scan.
Sampling

• Large databases
• Sample the database and apply Apriori to the sample.

• Potentially Large Itemsets (PL): Large itemsets from sample

• Negative Border (BD⁻):  
  – Generalization of Apriori-Gen applied to itemsets of varying sizes.
  – Minimal set of itemsets which are not in PL, but whose every subset is in PL.
Negative Border Example

PL

PL \cup B D^{-}(PL)
Sampling Algorithm

1. \( D_s = \) sample of Database \( D \);
2. \( PL = \) Large itemsets in \( D_s \) using \( \alpha \text{MinSup} \);
3. \( C = PL \cup BD^-(PL) \);
4. Count \( C \) in \( D_s \);
5. \( ML = \) large itemsets in \( BD^-(PL) \);
6. If \( ML = \emptyset \) then done
7. else \( C = \) repeated application of \( BD^- \);
8. Count \( C \) in Database;
Sampling Example

- Find AR assuming MinSup = 20%
- $D_s = \{ t_1, t_2 \}$
- $\alpha_{MinSup} = 10\%$
- $PL = \{ \{\text{Bread}\}, \{\text{Jelly}\}, \{\text{PeanutButter}\}, \{\text{Bread, Jelly}\}, \{\text{Bread, PeanutButter}\}, \{\text{Jelly, PeanutButter}\}, \{\text{Bread, Jelly, PeanutButter}\} \}$
- $BD^{-}(PL) = \{ \{\text{Beer}\}, \{\text{Milk}\}\}$
- $ML = \{ \{\text{Beer}\}, \{\text{Milk}\}\}$
- Repeated application of $BD^{-}$ generates all remaining itemsets
Sampling Adv/Disadv

• **Advantages:**
  – Reduces number of database scans to one in the best case and two in worst.
  – Scales better.

• **Disadvantages:**
  – Potentially large number of candidates in second pass
Parallelizing AR Algorithms

• Based on Apriori
• Techniques differ:
  – What is counted at each site
  – How data (transactions) are distributed
• Data Parallelism
  – Data partitioned
  – Count Distribution Algorithm
• Task Parallelism
  – Data and candidates partitioned
  – Data Distribution Algorithm
Count Distribution Algorithm (CDA)

1. Place data partition at each site.
2. In Parallel at each site do
3. \( C_1 = \) Itemsets of size one in \( I \);
4. Count \( C_1 \);
5. Broadcast counts to all sites;
6. Determine global large itemsets of size 1, \( L_1 \);
7. \( i = 1 \);
8. Repeat
9. \( i = i + 1 \);
10. \( C_i = \) Apriori-Gen\( (L_{i-1}) \);
11. Count \( C_i \);
12. Broadcast counts to all sites;
13. Determine global large itemsets of size \( i \), \( L_i \);
14. until no more large itemsets found;
CDA Example

$p^1$

$D^1$: $t_1, t_2$
Counts:
- Beer: 0
- Bread: 2
- Jelly: 1
- Milk: 0
- PeanutButter: 2

$p^2$

$D^2$: $t_3, t_4$
Counts:
- Beer: 1
- Bread: 2
- Jelly: 0
- Milk: 1
- PeanutButter: 1

$p^3$

$D^3$: $t_5$
Counts:
- Beer: 1
- Bread: 0
- Jelly: 0
- Milk: 1
- PeanutButter: 0

Broadcast Local Counts
Data Distribution Algorithm (DDA)

1. Place data partition at each site.
2. In Parallel at each site do
3. Determine local candidates of size 1 to count;
4. Broadcast local transactions to other sites;
5. Count local candidates of size 1 on all data;
6. Determine large itemsets of size 1 for local candidates;
7. Broadcast large itemsets to all sites;
8. Determine $L_1$;
9. $i = 1$;
10. Repeat
11. $i = i + 1$;
12. $C_i = \text{Apriori-Gen}(L_{i-1})$;
13. Determine local candidates of size $i$ to count;
14. Count, broadcast, and find $L_i$;
15. until no more large itemsets found;
**DDA Example**

Broadcast Database Partition

<table>
<thead>
<tr>
<th>P1</th>
<th>P2</th>
<th>P3</th>
</tr>
</thead>
</table>
| **D1:**  
  \[ t_1 , t_2 \]  
  Counts:  
  Beer 0  
  Bread 2 |
| **D2:**  
  \[ t_3 , t_4 \]  
  Counts:  
  Jelly 0  
  Milk 1 |
| **D3:**  
  \[ t_5 \]  
  Counts:  
  PeanutButter 0 |
## Comparison of AR Techniques

<table>
<thead>
<tr>
<th>Partitioning</th>
<th>Scans</th>
<th>Data Structure</th>
<th>Parallelism</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apriori</td>
<td>( m + 1 )</td>
<td>hash tree</td>
<td>none</td>
</tr>
<tr>
<td>Sampling</td>
<td>2</td>
<td>not specified</td>
<td>none</td>
</tr>
<tr>
<td>Partitioning</td>
<td>2</td>
<td>hash table</td>
<td>none</td>
</tr>
<tr>
<td>CDA</td>
<td>( m + 1 )</td>
<td>hash tree</td>
<td>data</td>
</tr>
<tr>
<td>DDA</td>
<td>( m + 1 )</td>
<td>hash tree</td>
<td>task</td>
</tr>
</tbody>
</table>
Incremental Association Rules

- Generate ARs in a dynamic database.
- Problem: algorithms assume static database
- Objective:
  - Know large itemsets for D
  - Find large itemsets for $D \cup \{\Delta D\}$
- Must be large in either D or $\Delta D$
- Save $L_i$ and counts
Note on ARs

• Many applications outside market basket data analysis
  – Prediction (telecom switch failure)
  – Web usage mining

• Many different types of association rules
  – Temporal
  – Spatial
  – Causal
Advanced AR Techniques

• Generalized Association Rules
  – Need is-a hierarchy
• Multiple-Level Association Rules
• Quantitative Association Rules
• Using multiple minimum supports

Figure 1: Example of a Taxonomy

Figure 1: A taxonomy for the relevant data items
Measuring Quality of Rules

- Support (Joint probability)
- Confidence (Conditional probability)
- Interest (Essentially a measure of independence)
- Conviction (Asymmetrical interest measure)
  - $A \rightarrow B$ rewritten using the implication elimination of P.L?
- Chi Squared Test
  - Create a contingency table, test for independence