Association Rules

Evgueni Smirnov
Overview

• Association Rule Problem
• Applications
• The Apriori Algorithm
• Discovering Association Rules
• Techniques to Improve Efficiency of Association Rule Mining
• Measures for Association Rules
Association Rule Problem

• Given a database of transactions:

<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>

• Find all the association rules:

<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>
Applications

• **Market Basket Analysis:** given a database of customer transactions, where each transaction is a set of items the goal is to find groups of items which are frequently purchased together.

• **Telecommunication** (each customer is a transaction containing the set of phone calls)

• **Credit Cards/ Banking Services** (each card/account is a transaction containing the set of customer’s payments)

• **Medical Treatments** (each patient is represented as a transaction containing the ordered set of diseases)

• **Basketball-Game Analysis** (each game is represented as a transaction containing the ordered set of ball passes)
Association Rule Definitions

- $I = \{i_1, i_2, ..., i_n\}$: a set of all the items
- Transaction $T$: a set of items such that $T \subseteq I$
- Transaction Database $D$: a set of transactions
- A transaction $T \subseteq I$ contains a set $X \subseteq I$ of some items, if $X \subseteq T$
- *An Association Rule*: is an implication of the form $X \Rightarrow Y$, where $X, Y \subseteq I$
Association Rule Definitions

• A set of items is referred as an itemset. A itemset that contains $k$ items is a $k$-itemset.
• The support $s$ of an itemset $X$ is the percentage of transactions in the transaction database $D$ that contain $X$.
• The support of the rule $X \Rightarrow Y$ in the transaction database $D$ is the support of the items set $X \cup Y$ in $D$.
• The confidence of the rule $X \Rightarrow Y$ in the transaction database $D$ is the ratio of the number of transactions in $D$ that contain $X \cup Y$ to the number of transactions that contain $X$ in $D$. 
Example

• Given a database of transactions:

<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>

• Find all the association rules:

<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 $I$ of all the items;
  — a database $D$ of transactions;
  — minimum support $s$;
  — minimum confidence $c$;

• Find:
  — all association rules $X \Rightarrow Y$ with a minimum support $s$ and confidence $c$. 
Problem Decomposition

1. Find all sets of items that have minimum support (frequent itemsets)
2. Use the frequent itemsets to generate the desired rules
If the minimum support is 50%, then \{Shoes, Jacket\} is the only 2-itemset that satisfies the minimum support.

If the minimum confidence is 50%, then the only two rules generated from this 2-itemset, that have confidence greater than 50%, are:

- **Shoes \(\Rightarrow\) Jacket**: Support=50%, Confidence=66%
- **Jacket \(\Rightarrow\) Shoes**: Support=50%, Confidence=100%
The Apriori Algorithm

• **Frequent Itemset Property:**
  Any subset of a frequent itemset is frequent.

• **Contrapositive:**
  If an itemset is not frequent, none of its supersets are frequent.
Frequent Itemset Property
The Apriori Algorithm

- $L_k$: Set of frequent itemsets of size $k$ (with min support)
- $C_k$: Set of candidate itemset of size $k$ (potentially frequent itemsets)

$L_1 = \text{frequent items};$

for $(k = 1; L_k \neq \emptyset; k++)$ do

$C_{k+1} = \text{candidates generated from } L_k;$

for each transaction $t$ in database do

increment the count of all candidates in $C_{k+1}$ that are contained in $t$

$L_{k+1} = \text{candidates in } C_{k+1} \text{ with min_support}$

return $\bigcup_k L_k;$
The Apriori Algorithm — Example

Min support = 50%

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>

C_1

Scan D

<table>
<thead>
<tr>
<th>itemset</th>
<th>sup.</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1}</td>
<td>2</td>
</tr>
<tr>
<td>{2}</td>
<td>3</td>
</tr>
<tr>
<td>{3}</td>
<td>3</td>
</tr>
<tr>
<td>{4}</td>
<td>1</td>
</tr>
<tr>
<td>{5}</td>
<td>3</td>
</tr>
</tbody>
</table>

L_1

L_2

Scan D

<table>
<thead>
<tr>
<th>itemset</th>
<th>sup.</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1 2}</td>
<td>1</td>
</tr>
<tr>
<td>{1 5}</td>
<td>1</td>
</tr>
</tbody>
</table>

C_2

Scan D

<table>
<thead>
<tr>
<th>itemset</th>
<th>sup.</th>
</tr>
</thead>
<tbody>
<tr>
<td>{1}</td>
<td>2</td>
</tr>
<tr>
<td>{2}</td>
<td>3</td>
</tr>
<tr>
<td>{3}</td>
<td>3</td>
</tr>
<tr>
<td>{5}</td>
<td>3</td>
</tr>
</tbody>
</table>

C_3

Scan D

L_3

<table>
<thead>
<tr>
<th>itemset</th>
<th>sup.</th>
</tr>
</thead>
<tbody>
<tr>
<td>{2 3 5}</td>
<td>2</td>
</tr>
</tbody>
</table>
How to Generate Candidates

**Input:** \( L_{i-1} \): set of frequent itemsets of size \( i-1 \)

**Output:** \( C_i \): set of candidate itemsets of size \( i \)

\[ C_i = \text{empty set}; \]

for each itemset \( J \) in \( L_{i-1} \) do

    for each itemset \( K \) in \( L_{i-1} \) s.t. \( K<> J \) do

        if \( i-2 \) of the elements in \( J \) and \( K \) are equal then

            if all subsets of \( \{ K \cup J \} \) are in \( L_{i-1} \) then

                \[ C_i = C_i \cup \{ K \cup J \} \]

return \( C_i \);
Example of Generating Candidates

• $L_3 = \{abc, abd, acd, ace, bcd\}$

• Generating $C_4$ from $L_3$
  – $abcd$ from $abc$ and $abd$
  – $acde$ from $acd$ and $ace$

• Pruning:
  – $acde$ is removed because $ade$ is not in $L_3$

• $C_4 = \{abcd\}$
Example of Discovering Rules

Let us consider the 3-itemset \{I1, I2, I5\}:

\[ \begin{align*}
I1 \land I2 & \Rightarrow I5 \\
I1 \land I5 & \Rightarrow I2 \\
I2 \land I5 & \Rightarrow I1 \\
I1 & \Rightarrow I2 \land I5 \\
I2 & \Rightarrow I1 \land I5 \\
I5 & \Rightarrow I1 \land I2
\end{align*} \]
Discovering Rules

for each frequent itemset \( I \) do

for each subset \( C \) of \( I \) do

if \( \frac{\text{support}(I)}{\text{support}(I - C)} \geq \text{minconf} \) then

output the rule \( (I - C) \Rightarrow C \),

with confidence = \( \frac{\text{support}(I)}{\text{support}(I - C)} \)

and support = \( \text{support}(I) \)
### Example of Discovering Rules

<table>
<thead>
<tr>
<th>TID</th>
<th>List of Item_IDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>T100</td>
<td>I1, I2, I5</td>
</tr>
<tr>
<td>T200</td>
<td>I2, I4</td>
</tr>
<tr>
<td>T300</td>
<td>I2, I3</td>
</tr>
<tr>
<td>T400</td>
<td>I1, I2, I4</td>
</tr>
<tr>
<td>T500</td>
<td>I1, I3</td>
</tr>
<tr>
<td>T600</td>
<td>I2, I3</td>
</tr>
<tr>
<td>T700</td>
<td>I1, I3</td>
</tr>
<tr>
<td>T800</td>
<td>I1, I2, I3, I5</td>
</tr>
<tr>
<td>T900</td>
<td>I1, I2, I3</td>
</tr>
</tbody>
</table>

Let us consider the 3-itemset \{I1, I2, I5\} with support of 0.22(2)%.

Let generate all the association rules from this itemset:

\[ I1 \land I2 \Rightarrow I5 \] \hspace{1cm} confidence= \frac{2}{4} = 50\%

\[ I1 \land I5 \Rightarrow I2 \] \hspace{1cm} confidence= \frac{2}{2} = 100\%

\[ I2 \land I5 \Rightarrow I1 \] \hspace{1cm} confidence= \frac{2}{2} = 100\%

\[ I1 \Rightarrow I2 \land I5 \] \hspace{1cm} confidence= \frac{2}{6} = 33\%

\[ I2 \Rightarrow I1 \land I5 \] \hspace{1cm} confidence= \frac{2}{7} = 29\%

\[ I5 \Rightarrow I1 \land I2 \] \hspace{1cm} confidence= \frac{2}{2} = 100\%
Apriori Advantages/Disadvantages

• **Advantages:**
  – Uses large itemset property.
  – Easily parallelized
  – Easy to implement.

• **Disadvantages:**
  – Assumes transaction database is memory resident.
  – Requires many database scans.
Transaction reduction

A transaction that does not contain any frequent $k$-itemset will not contain frequent $l$-itemset for $l > k$! Thus, it is useless in subsequent scans!
Partitioning

Transactions in D → Divided D into partitions → Find the frequent itemsets local to each partition (1 scan) → Combine all local frequent itemsets to form candidate itemset → Find global frequent itemsets among candidates (1 scan) → Frequent itemsets in D
Sampling

Mining on a subset of given data, lower support threshold + a method to determine the completeness
Alternative Measures for Association Rules

• The **confidence** of \( X \Rightarrow Y \) in database \( D \) is the ratio of the number of transactions containing \( X \cup Y \) to the number of transactions that contain \( X \). In other words the confidence is:

\[
\text{conf}(X \rightarrow Y) = \frac{|D|}{\text{numTrans}(X)} = \frac{p(X \land Y)}{p(X)} = p(Y \mid X)
\]

• But, when \( Y \) is independent of \( X \): \( p(Y) = p(Y \mid X) \). In this case if \( p(Y) \) is high we’ll have a rule with high confidence that associate independent itemsets! For example, if \( p(\text{“buy milk”}) = 80\% \) and “buy milk” is independent from “buy salmon”, then the rule “buy salmon” \( \Rightarrow \) “buy milk” will have confidence 80%!
Alternative Measures for Association Rules

• The lift measure indicates the departure from independence of $X$ and $Y$. The lift of $X \Rightarrow Y$ is:

$$
lift(X \rightarrow Y) = \frac{\text{conf}(X \rightarrow Y)}{p(Y)} = \frac{p(X \land Y)}{p(X)} = \frac{p(X \land Y)}{p(X)p(Y)}
$$

• But, the lift measure is symmetric; i.e., it does not take into account the direction of implications!
Alternative Measures for Association Rules

The **conviction** measure indicates the departure from independence of $X$ and $Y$ taking into account the implication direction. The conviction of $X \Rightarrow Y$ is:

\[
\text{conv}(X \rightarrow Y) = \frac{p(X)p(\neg Y)}{p(X \land \neg Y)}
\]
Summary

1. Association Rules form an very applied data mining approach.

2. Association Rules are derived from frequent itemsets.

3. The Apriori algorithm is an efficient algorithm for finding all frequent itemsets.

4. The Apriori algorithm implements level-wise search using frequent item property.

5. The Apriori algorithm can be additionally optimised.

6. There are many measures for association rules.