Association Rules

presented by

Zbigniew W. Ras*,#)

*) University of North Carolina - Charlotte
#) Polish-Japanese Academy of Information Technology
Market Basket Analysis (MBA)

Customer buying habits by finding associations and correlations between the different items that customers place in their “shopping basket”

Customer1: Milk, eggs, sugar, bread

Customer2: Milk, eggs, cereal, bread

Customer3: Eggs, sugar
Market Basket Analysis

Given: a database of customer transactions, where each transaction is a set of items

Find groups of items which are frequently purchased together

![Diagram showing a transaction of items A, B, and C on a receipt.](image)
Goal of MBA

- Extract information on purchasing behavior

- Actionable information: can suggest
  - new store layouts
  - new product assortments
  - which products to put on promotion

MBA applicable whenever a customer purchases multiple things in proximity
Association Rules

- Express how product/services relate to each other, and tend to group together

- “if a customer purchases three-way calling, then will also purchase call-waiting”

- Simple to understand

- Actionable information: bundle three-way calling and call-waiting in a single package
Basic Concepts

Transactions:
- Relational format
  - \(<\text{Tid}, \text{item}>\)
  - \(<1, \text{item1}>\)
  - \(<1, \text{item2}>\)
  - \(<2, \text{item3}>\)
- Compact format
  - \(<\text{Tid}, \text{itemset}>\)
  - \(<1, \{\text{item1, item2}\}>\)
  - \(<2, \{\text{item3}\}>\)

**Item:** single element, **Itemset:** set of items

**Support of an itemset** \(I\) [denoted by \(\text{sup}(I)\)]: \(\text{card}(I)\)

Threshold for minimum support: \(\sigma\)

**Itemset** \(I\) is **Frequent** if: \(\text{sup}(I) \geq \sigma\).

Frequent Itemset represents set of items which are positively correlated
Frequent Itemsets

<table>
<thead>
<tr>
<th>Transaction ID</th>
<th>Items Bought</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>dairy, fruit</td>
</tr>
<tr>
<td>2</td>
<td>dairy, fruit, vegetable</td>
</tr>
<tr>
<td>3</td>
<td>dairy</td>
</tr>
<tr>
<td>4</td>
<td>fruit, cereals</td>
</tr>
</tbody>
</table>

$\text{sup}(\{\text{dairy}\}) = 3$

$\text{sup}(\{\text{fruit}\}) = 3$

$\text{sup}(\{\text{dairy, fruit}\}) = 2$

If $\sigma = 3$, then

$\{\text{dairy}\}$ and $\{\text{fruit}\}$ are frequent while $\{\text{dairy, fruit}\}$ is not.
Association Rules: $\text{AR}(s,c)$

- $\{A,B\}$ - partition of a set of items
- $r = [A \Rightarrow B]$
  
  **Support of** $r$: $\text{sup}(r) = \text{sup}(A \cup B)$
  
  **Confidence of** $r$: $\text{conf}(r) = \frac{\text{sup}(A \cup B)}{\text{sup}(A)}$

- **Thresholds:**
  - minimum support - $s$
  - minimum confidence - $c$

  $r \in \text{AS}(s, c)$, if $\text{sup}(r) \geq s$ and $\text{conf}(r) \geq c$
For rule $A \Rightarrow C$:

- $\text{sup}(A \Rightarrow C) = 2$
- $\text{conf}(A \Rightarrow C) = \frac{\text{sup}\{A,C\}}{\text{sup}\{A\}} = \frac{2}{3}$

The Apriori principle:

- Any subset of a frequent itemset must be frequent
The Apriori algorithm [Agrawal]

- $F_k$: Set of frequent itemsets of size $k$
- $C_k$: Set of candidate itemsets of size $k$

\[ F_1 := \{\text{frequent items}\}; \ k := 1; \]
\[ \text{while } \mathrm{card}(F_k) \geq 1 \ \text{do} \begin{align*}
C_{k+1} &:= \text{new candidates generated from } F_k; \\
\text{for each transaction } t \ \text{in the database} \ &\text{do} \\
&\text{increment the count of all candidates in } C_{k+1} \ \text{that} \\
&\text{are contained in } t; \\
F_{k+1} &:= \text{candidates in } C_{k+1} \ \text{with minimum support} \\
k &:= k+1 \\
\end{align*} \]
\[ \text{end} \]

Answer := $\cup \{ F_k: k \geq 1 \ \& \ \mathrm{card}(F_k) \geq 1 \}$
{a,d} is not frequent, so the 3-itemsets \{a,b,d\}, \{a,c,d\} and the 4-itemset \{a,b,c,d\}, are not generated.
Representative Association Rules

Definition 1. Cover \( C \) of a rule \( X \Rightarrow Y \) is denoted by \( C(X \Rightarrow Y) \) and defined as follows:
\[
C(X \Rightarrow Y) = \{ [X \cup Z] \Rightarrow V : Z, V \text{ are disjoint subsets of } Y \}.
\]

Definition 2. Set \( RR(s, c) \) of Representative Association Rules is defined as follows:
\[
RR(s, c) = \{ r \in AR(s, c) : \neg(\exists r_1 \in AR(s, c)) [r_1 \neq r \& r \in C(r_1)]\}
\]
s - threshold for minimum support
c - threshold for minimum confidence

Representative Rules (informal description):
[as short as possible] \( \Rightarrow \) [as long as possible]
**Representative Association Rules**

Transactions:
- \{A,B,C,D,E\}
- \{A,B,C,D,E,F\}
- \{A,B,C,D,E,H,I\}
- \{A,B,E\}
- \{B,C,D,E,H,I\}

Find RR(2,80%) 

Representative Rules

From (BCDEHI):
- \{H\} ⇒ \{B,C,D,E,I\}
- \{I\} ⇒ \{B,C,D,E,H\}

From (ABCDE):
- \{A,C\} ⇒ \{B,D,E\}
- \{A,D\} ⇒ \{B,C,E\}

<table>
<thead>
<tr>
<th>1-element sets</th>
<th>2-element sets</th>
<th>3-element sets</th>
<th>4-element sets</th>
<th>5-element sets</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A, 4)</td>
<td>(AB, 4)</td>
<td>(ABC, 3)</td>
<td>(ABCD, 3)</td>
<td>(ABCDE, 3)</td>
</tr>
<tr>
<td>(B, 5)</td>
<td>(AC, 3)</td>
<td>(ABD, 3)</td>
<td>(ABCE, 3)</td>
<td></td>
</tr>
<tr>
<td>(C, 4)</td>
<td>(AD, 3)</td>
<td>(ABE, 4)</td>
<td>(ABDE, 3)</td>
<td></td>
</tr>
<tr>
<td>(D, 4)</td>
<td>(AE, 4)</td>
<td>(ACD, 3)</td>
<td>(ACDE, 3)</td>
<td></td>
</tr>
<tr>
<td>(E, 5)</td>
<td>(BC, 4)</td>
<td>(ACE, 3)</td>
<td>(ACF, 3)</td>
<td></td>
</tr>
<tr>
<td>(H, 2)</td>
<td>(BD, 4)</td>
<td>(ADE, 3)</td>
<td>(ADH, 3)</td>
<td></td>
</tr>
<tr>
<td>(I, 2)</td>
<td>(BE, 5)</td>
<td>(BCD, 4)</td>
<td>(BCDE, 4)</td>
<td>(BCDEH, 2)</td>
</tr>
<tr>
<td>(BH, 2)</td>
<td>(BCE, 4)</td>
<td>(BCDH, 2)</td>
<td>(BCDEI, 2)</td>
<td></td>
</tr>
<tr>
<td>(BI, 2)</td>
<td>(BCI, 2)</td>
<td>(BCDI, 2)</td>
<td>(BCDEH, 2)</td>
<td></td>
</tr>
<tr>
<td>(CD, 4)</td>
<td>(BCI, 2)</td>
<td>(BCDI, 2)</td>
<td>(BCDEH, 2)</td>
<td></td>
</tr>
<tr>
<td>(CE, 4)</td>
<td>(BDE, 4)</td>
<td>(BCEI, 2)</td>
<td>(BDEH, 2)</td>
<td></td>
</tr>
<tr>
<td>(CH, 2)</td>
<td>(BDH, 2)</td>
<td>(BCHI, 2)</td>
<td>(BDEH, 2)</td>
<td></td>
</tr>
<tr>
<td>(CI, 2)</td>
<td>(BDI, 2)</td>
<td>(BDEH, 2)</td>
<td>(BDEH, 2)</td>
<td></td>
</tr>
<tr>
<td>(DE, 4)</td>
<td>(BEH, 2)</td>
<td>(BDEI, 2)</td>
<td>(BDEH, 2)</td>
<td></td>
</tr>
<tr>
<td>(DI, 2)</td>
<td>(BHI, 2)</td>
<td>(BDEH, 2)</td>
<td>(BDEH, 2)</td>
<td></td>
</tr>
<tr>
<td>(EI, 2)</td>
<td>(CDH, 2)</td>
<td>(CDEI, 2)</td>
<td>(CDEH, 2)</td>
<td></td>
</tr>
<tr>
<td>(HI, 2)</td>
<td>(CDI, 2)</td>
<td>(CDHI, 2)</td>
<td>(CDEH, 2)</td>
<td>(CDEHI, 2)</td>
</tr>
<tr>
<td>(CEH, 2)</td>
<td></td>
<td>(CEHI, 2)</td>
<td>(CEHI, 2)</td>
<td></td>
</tr>
<tr>
<td>(CHI, 2)</td>
<td></td>
<td></td>
<td>(DEHI, 2)</td>
<td></td>
</tr>
<tr>
<td>(DEH, 2)</td>
<td></td>
<td></td>
<td>(DEHI, 2)</td>
<td></td>
</tr>
<tr>
<td>(DEI, 2)</td>
<td></td>
<td></td>
<td>(DEHI, 2)</td>
<td></td>
</tr>
<tr>
<td>(DHI, 2)</td>
<td></td>
<td></td>
<td>(DEHI, 2)</td>
<td></td>
</tr>
<tr>
<td>(EHI, 2)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Last set: (BCDEHI, 2)
Frequent Pattern (FP) Growth Strategy

Transactions:
abcde
abc
acde
bcde
bc
bde
cde

Transactions ordered:
cbdea
cba
cdea
cbde
cb
bde
cde

Frequent Items:
c – 6
b – 5
d – 5
e – 5
a – 3

Minimum Support = 2

FP-tree
Frequent Pattern (FP) Growth Strategy

Mining the FP-tree for frequent itemsets:
Start from each item and construct a subdatabase of transactions (prefix paths) with that item listed at the end.
Reorder the prefix paths in support descending order. Build a conditional FP-tree.

Prefix path:
(c b d e a, 1)
(c b a, 1)
(c d e a, 1)

Correct order:
c – 3
b – 2
d – 2
e – 2
a – 3
Frequent Pattern (FP) Growth Strategy

Prefix path:
- (c b d e a, 1)
- (c b a, 1)
- (c d e a, 1)

Frequent Itemsets:
- (c a, 3)
- (c b a, 2)
- (c d a, 2)
- (c d e a, 2)
- (c e a, 2)
Multidimensional AR

Associations between values of different attributes:

<table>
<thead>
<tr>
<th>CID</th>
<th>nationality</th>
<th>age</th>
<th>income</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Italian</td>
<td>50</td>
<td>low</td>
</tr>
<tr>
<td>2</td>
<td>French</td>
<td>40</td>
<td>high</td>
</tr>
<tr>
<td>3</td>
<td>French</td>
<td>30</td>
<td>high</td>
</tr>
<tr>
<td>4</td>
<td>Italian</td>
<td>50</td>
<td>medium</td>
</tr>
<tr>
<td>5</td>
<td>Italian</td>
<td>45</td>
<td>high</td>
</tr>
<tr>
<td>6</td>
<td>French</td>
<td>35</td>
<td>high</td>
</tr>
</tbody>
</table>

RULES:

[nationality = French] ⇒ [income = high]  [50%, 100%]
[income = high] ⇒ [nationality = French]  [50%, 75%]
[age = 50] ⇒ [nationality = Italian]      [33%, 100%]
### Single-dimensional AR vs Multi-dimensional

<table>
<thead>
<tr>
<th>Multi-dimensional</th>
<th>Single-dimensional</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;1, Italian, 50, low&gt;</td>
<td>&lt;1, {nat/Ita, age/50, inc/low}&gt;</td>
</tr>
<tr>
<td>&lt;2, French, 45, high&gt;</td>
<td>&lt;2, {nat/Fre, age/45, inc/high}&gt;</td>
</tr>
</tbody>
</table>

Schema: <ID, a?, b?, c?, d?>

<table>
<thead>
<tr>
<th>Multi-dimensional</th>
<th>Single-dimensional</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;1, yes, yes, no, no&gt;</td>
<td>&lt;1, {a, b}&gt;</td>
</tr>
<tr>
<td>&lt;2, yes, no, yes, no&gt;</td>
<td>&lt;2, {a, c}&gt;</td>
</tr>
</tbody>
</table>


Quantitative Attributes

- Quantitative attributes (e.g. age, income)
- Categorical attributes (e.g. color of car)

<table>
<thead>
<tr>
<th>CID</th>
<th>height</th>
<th>weight</th>
<th>income</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>168</td>
<td>75.4</td>
<td>30.5</td>
</tr>
<tr>
<td>2</td>
<td>175</td>
<td>80.0</td>
<td>20.3</td>
</tr>
<tr>
<td>3</td>
<td>174</td>
<td>70.3</td>
<td>25.8</td>
</tr>
<tr>
<td>4</td>
<td>170</td>
<td>65.2</td>
<td>27.0</td>
</tr>
</tbody>
</table>

Problem: too many distinct values

Solution: transform quantitative attributes into categorical ones via discretization.
Discretization of quantitative attributes

- Quantitative attributes are **statically** discretized by using predefined concept hierarchies:
  - elementary use of background knowledge

Loose interaction between Apriori and Discretizer

- Quantitative attributes are **dynamically** discretized
  - into “bins” based on the distribution of the data.
  - considering the distance between data points.

Tighter interaction between Apriori and Discretizer
Constraint-based AR

- **Preprocessing:** use constraints to focus on a subset of transactions
  - Example: find association rules where the prices of all items are at most 200 Euro

- **Optimizations:** use constraints to optimize Apriori algorithm
  - Anti-monotonicity: when a set violates the constraint, so does any of its supersets.
  - Apriori algorithm uses this property for pruning

- **Push constraints as deep as possible inside the frequent set computation**
Anti-monotonicity: If a set $S$ violates the constraint, any superset of $S$ violates the constraint.

Examples:
- $[\text{Price}(S) \leq v]$ is anti-monotone
- $[\text{Price}(S) \geq v]$ is not anti-monotone
- $[\text{Price}(S) = v]$ is partly anti-monotone

Application:
- Push $[\text{Price}(S) \leq 1000]$ deeply into iterative frequent set computation.
Post processing

- A naive solution: apply Apriori for finding all frequent sets, and then to test them for constraint satisfaction one by one.

Optimization

- Han’s approach: comprehensive analysis of the properties of constraints and try to push them as deeply as possible inside the frequent set computation.
Mining Multilevel AR

Hierarchical attributes: age, salary

Association Rule: \((age, \text{young}) \rightarrow (salary, 40k)\)

Candidate Association Rules:
\((age, 18) \rightarrow (salary, 40k), \quad (age, \text{young}) \rightarrow (salary, \text{low}), \quad (age, 18) \rightarrow (salary, \text{low})\)
Questions?

Thank You