Market Basket Analysis - I

Contents

About Association Rules

Association Rule Learning

Method for discovering interesting relations between variables in large databases

- Based on the **concept of strong rules**, Rakesh Agrawal introduced association rules for **discovering regularities between products in large-scale transaction data recorded by point-of-sale (POS) systems in supermarkets**
- For example, the rule found in the sales data of a supermarket would indicate that **if a customer buys onions and potatoes together, they are also likely to buy burger**
- Association rule learning method can be applied in many areas such as **web usage mining, fraud detection, continuous production and bioinformatics**

Introduction to Market Basket Analysis

- The most widely used area of application for association rules is **Market Basket Analysis**

> Market Basket Analysis (Association Analysis) is a **mathematical modeling technique based upon the theory that if you buy a certain group of items, you are likely to buy another group of items**

- It is used to **analyze the customer purchasing behavior and helps in increasing the sales and maintain inventory** by focusing on the point of sale transaction data

Market Basket Analysis – Uses

Product Building	• Develop combo offers based on products bought together
Optimisation	• Organise and place associated products/categories nearby inside a store
Advertising and Marketing	• Determine the layout of the catalog of an ecommerce site
Inventory Management	• Control inventory based on product demands and what products sell together

Definitions and Terminology

Term	Definition
Transactions	A set of items (Item set)
Support	Ratio of number of times two or more items occur together to the total number of transactions Support can be thought of as P(A and B)
Confidence	Conditional probability that a randomly selected transaction will include Item B given Item A P(B|A) (written as A => B)
Lift	Ratio of the probability of Items A and B occurring together (Joint probability) to the product of P(A) and P(B)

Get an Edge!

The Famous Story

> An article in The Financial Times of London (Feb. 7, 1996) stated,
>
> "The example of what data mining can achieve is the case of a large US supermarket chain which discovered a strong association for many customers between a brand of babies nappies (diapers) and a brand of beer. Most customers who bought the nappies also bought the beer. The best hypothesisers in the world would find it difficult to propose this combination but data mining showed it existed, and the retail outlet was able to exploit it by moving the products closer together on the shelves."

Rule Evaluation – Support

Transaction No.	Item 1	Item 2	Item 3	...
100	Beer	Diaper	Chocolate	
101	Milk	Chocolate	Shampoo	
102	Beer	Wine	Vodka	
103	Beer	Cheese	Diaper	
104	Ice Cream	Diaper	Beer	

A B

Support of {Diaper, Beer}

$$\text{Support} = \frac{\text{No.of transactions containing both A and B}}{\text{Total no.of transactions}} = \frac{3}{5} = 60\%$$

Support of {Diaper, Beer} is 3/5

Rule Evaluation – Confidence

Transaction No.	Item 1	Item 2	Item 3	...
100	Beer	Diaper	Chocolate	
101	Milk	Chocolate	Shampoo	
102	Beer	Wine	Vodka	
103	Beer	Cheese	Diaper	
104	Ice Cream	Diaper	Beer	

$$\text{Confidence for } \{A\} \Rightarrow \{B\} = \frac{\text{No. of transactions containing both A and B}}{\text{No. of transactions containing A}}$$

Confidence for {Diaper} ⇒ {Beer} is 3/3

When Diaper is purchased, the likelihood of Beer purchase is 100%

Confidence for {Beer} ⇒ {Diaper} is 3/4

When Beer is purchased, the likelihood of Diaper purchase is 75%

{Diaper} ⇒ {Beer} is a more important rule according to Confidence

Rule Evaluation – Lift

Transaction No.	Item 1	Item 2	Item 3	Item 4
100	Beer	Diaper	Chocolate	
101	Milk	Chocolate	Shampoo	
102	Beer	Milk	Vodka	Chocolate
103	Beer	Milk	Diaper	Chocolate
104	Milk	Diaper	Beer	

A B

Consider {Chocolate} ⇒ {Milk}

$$\text{Lift} = \frac{P(A \cap B)}{P(A)P(B)} = \frac{3/5}{(4/5)(4/5)} = 0.9375$$

Lift < 1 indicates Chocolate is decreasing the chance of Milk purchase
Support and confidence are high but lift is low

Case Study – Groceries Purchase Data

Background

- A typical grocery outlet records point-of-sale transaction data

Objective

- To mine association rules and information about item sets

Available Information

- Total number of transactions is 9835
- Items are aggregated to 169 categories
- Data is collected for 1 month (30-days)

Data Snapshot

Groceries

```
    items
[1] {citrus fruit,
     semi-finished bread,
     margarine,
     ready soups}
[2] {tropical fruit,
     yogurt,
     coffee}
[3] {whole milk}
[4] {pip fruit,
     yogurt,
     cream cheese ,
     meat spreads}
[5] {other vegetables,
```

Columns	Description	Possible values
id	Transaction Id	Positive Integers
items	Set of Items purchased in a transaction	Subset from 169 categories of items

Market Basket Analysis in R

```
#Market Basket Analysis Using Apriori Recommendation

install.packages("arules")
library(arules)

install.packages("arulesViz")
library(arulesViz)


data("Groceries")
```

We will be using two packages for performing Market Basket Analysis in R.
Package **"arules"** stands for 'Association Rules' and it contains functions for mining association rules and frequent itemsets.
Package **"arulesViz"** is used for visualisation.
Install and load these two packages.

- ❑ Load the dataset.
- ❑ The **Groceries** data set is provided for package **arules** by Michael Hahsler, Kurt Hornik and Thomas Reutterer.*

The data is of class 'transaction' supported by package **arules**.

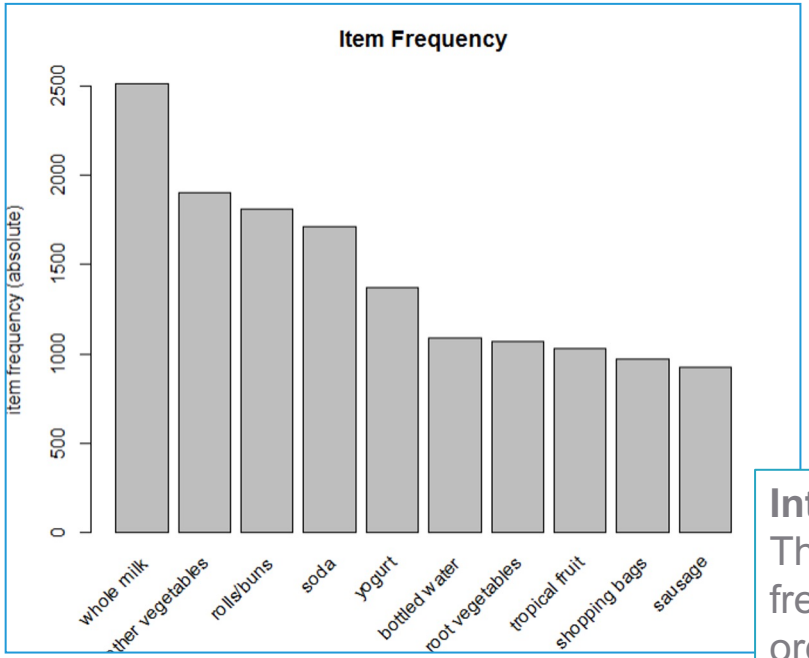
Visualise Item Frequency

```
#Item Frequency Plot
itemFrequencyPlot(Groceries,topN=10,type="absolute",
                  main="Item Frequency")
```

itemFrequencyPlot() calculates item frequency and returns a barplot.
topN= instructs R to plot only top N highest item frequency or lift (Logical, if **lift=TRUE**). It plots values in decreasing order.
type= is a character string indicating whether item frequencies should be displayed relative or absolute. Default is relative.

Item Frequency Plot

```
# Output
```

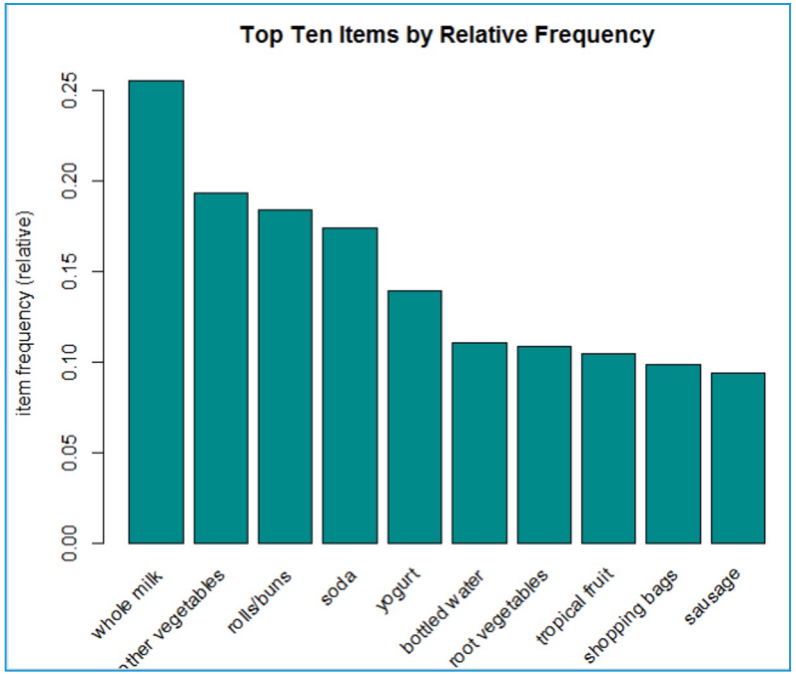


Interpretation:
The plot shows items by frequency in a descending order.

Item Frequency Plot

```
itemFrequencyPlot(Groceries,topN=10,type="relative",
col="darkcyan",main="Top Ten Items by Relative Frequency")
```

```
# Output
```



- **type= "relative"** displays barplot with the relative frequency
- **col=** specifies the colour of the bars

Interpretation:

- The plot shows items by relative frequency in a descending order.

Get and Display the Rules

```
#Get the Rules
rules<-apriori(Groceries,parameter=list(supp=0.001,conf=0.8))
```

- The Apriori algorithm employs level-wise search for frequent itemsets.
- **apriori()** is used to mine frequent itemsets, association rules or association hyperedges using this algorithm.
- The default is to mine rules with **support 0.1, confidence 0.8.**
- **Here, we have used threshold of 0.001 for support.**
- **apriori()** returns an object of class rules or itemsets.

```
#Show Top 5 Rules But Only 2 Digits
options(digits=2)
```

options in base R allows the user to set global options which affect the way in which R computes and displays results. We have set **digits=2** to display results with only 2 digits.

```
inspect(rules[1:5])
```

inspect in package **arules** displays association and plus additional information formatted for online inspection.

Get and Display the Rules

```
# Output of Rules
```

```
Apriori

Parameter specification:
 confidence minval smax arem  aval originalSupport maxtime support minlen
        0.8    0.1    1 none FALSE            TRUE       5   0.001      1
 maxlen target   ext
     10  rules FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE    2    TRUE

Absolute minimum support count: 9

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[169 item(s), 9835 transaction(s)] done [0.01s].
sorting and recoding items ... [157 item(s)] done [0.00s].
creating transaction tree ... done [0.00s].
checking subsets of size 1 2 3 4 5 6 done [0.03s].
writing ... [410 rule(s)] done [0.00s].
creating S4 object  ... done [0.00s].
```

Interpretation:

- The output displays parameter specification, algorithmic control and absolute minimum support count.
- It also lists down tasks performed and time taken to complete them.
- We are interested in knowing how many rules were created; 410 in our case.

Get and Display the Rules

```
# Output of inspect
```

	lhs		rhs	support	confidence	lift	count
[1]	{liquor,red/blush wine}	=>	{bottled beer}	0.0019	0.90	11.2	19
[2]	{curd,cereals}	=>	{whole milk}	0.0010	0.91	3.6	10
[3]	{yogurt,cereals}	=>	{whole milk}	0.0017	0.81	3.2	17
[4]	{butter,jam}	=>	{whole milk}	0.0010	0.83	3.3	10
[5]	{soups,bottled beer}	=>	{whole milk}	0.0011	0.92	3.6	11

Interpretation:

- **inspect()** returns list of lhs and rhs items, their support, confidence and lift values.

Manage How the Rules are Displayed

```
#Sort the Rules
rules<-sort(rules,by="lift",decreasing=TRUE)
```

- sort() from package arules is used
- by="lift" indicates sort by values of Lift
- decreasing= logical, specifies the direction of sorting. Default is decreasing=TRUE.

```
#Show Top 5 Rules (Sorted)

options(digits=2)

inspect(rules[1:5])
```

Top Five Rules (Sorted)

```
# Output
```

```
    lhs                                rhs                 support confidence lift count
[1] {liquor,
     red/blush wine}                => {bottled beer}      0.0019       0.90 11.2    19
[2] {citrus fruit,
     other vegetables,
     soda,
     fruit/vegetable juice}         => {root vegetables}   0.0010       0.91  8.3    10
[3] {tropical fruit,
     other vegetables,
     whole milk,
     yogurt,
     oil}                           => {root vegetables}   0.0010       0.91  8.3    10
[4] {citrus fruit,
     grapes,
     fruit/vegetable juice}         => {tropical fruit}    0.0011       0.85  8.1    11
[5] {other vegetables,
     whole milk,
     yogurt,
     rice}                          => {root vegetables}   0.0013       0.87  8.0    13
```

Interpretation:

- The rules are now sorted based on lift. Sorting ensures that most relevant rules appear first.

Quick Recap

In this session, we learnt Market Basket Analysis:

Market Basket Analysis

- Mathematical modeling technique based upon the theory that if you buy a certain group of items, you are likely to buy another group of items
- Transactions, Support, Confidence and Lift are the key concepts used in this analysis
- The analysis is performed by creating and studying rules based on different itemsets

Market Basket Analysis in R

- Package **`arules`** and **`arulesViz`** are used for undertaking MBA
- **`itemFrequencyPlot()`** plots frequency
- **`apriori()`** function creates rules. **`inspect()`** displays association and additional information
- **`plot()`** in **`arulesViz`** can create static or interactive plots