Market Basket Analysis

Market basket analysis (MBA) is a data mining technique that is used to uncover purchase patterns in any retail setting. MBA is a set of statistical affinity calculations that help business leaders better understand – and ultimately serve – their customers by highlighting purchasing patterns. In simplest terms, MBA looks for what combinations of products most frequently occur together in transactions. These relationships can be used to increase profitability through cross-selling, recommendations, promotions, or even the placement of items on a menu or in a store

Introduction

Market Basket Analysis is one of the key techniques used by large retailers to uncover associations between items. It works by looking for combinations of items that occur together frequently in transactions. To put it another way, it allows retailers to identify relationships between the items that people buy.

Association Rules are widely used to analyze retail basket or transaction data, and are intended to identify strong rules discovered in transaction data using measures of interestingness, based on the concept of strong rules.

An example of Association Rules

Assume there are 100 customers

10 of them bought milk, 8 bought butter and 6 bought both of them.

Bought milk => bought butter

Support = P(Milk & Butter) = 6/100 = 0.06

Confidence = support/P(Butter) = 0.06/0.08 = 0.75

Lift = confidence/P(Milk) = 0.75/0.10 = 7.5

This example is extremely small. In practice, a rule needs the support of several hundred transactions, before it can be considered statistically significant, and datasets often contain thousands or millions of transactions.

Load the packages

```
Library(tidyverse)

Library(readxl)

Library(knitr)

Library(ggplot2)

Library(lubridate)

Library(arules)

Library(arulesViz)

Library(plyr)
```

Data preprocessing and exploring

```
Retail <- read_excel(‘Online_retail.xlsx’)

Retail <- retail[complete.cases(retail), ]

Retail <- retail %>% mutate(Description = as.factor(Description))

Retail <- retail %>% mutate(Country = as.factor(Country))

Retail$Date <- as.Date(retail$InvoiceDate)

Retail$Time <- format(retail$InvoiceDate,”%H:%M:%S”)

Retail$InvoiceNo <- as.numeric(as.character(retail$InvoiceNo))

Glimpse(retail)
```

What time do people often purchase online?

In order to find the answer to this question, we need to extract “hour” from the time column.

```
retail$Time <-

as.factor(retail$Time) a <-

hms(as.character(retail$Time))

retail$Time = hour(a)
```

```
retail %>%    ggplot(aes(x=Time)) +
 geom_histogram(stat="count",fill="indianred")
```

How many items each customer buy?

```
Detach(“package:plyr”, unload=TRUE)
Retail %>%
  Group_by(InvoiceNo) %>%
  Summarize(n_items = mean(Quantity)) %>%
  Ggplot(aes(x=n_items))+
  Geom_histogram(fill=”indianred”, bins = 100000) +
  Geom_rug()+
```

Top 10 best sellers

```
Tmp <- retail %>%
  Group_by(StockCode, Description) %>%
  Summarize(count = n()) %>%
  Arrange(desc(count))
Tmp <- head(tmp, n=10)
Tmp
Tmp %>%
  Ggplot(aes(x=reorder(Description,count), y=count))+
  Geom_bar(stat=”identity”,fill=”indian red”)+
  Coord_flip()
```

Association rules for online retailer

Before using any rule mining algorithm, we need to transform the data from the data frame format, into transactions such that we have all the items bought together in one row.

```
Retail_sorted <- retail[order(retail$CustomerID),]
Library(plyr)
itemList <- ddply(retail,c(“CustomerID”,”Date”),
                function(df1)paste(df1$Description,
                collapse = “,”))
```

The function ddply() accepts a data frame, splits it into pieces based on one or more factors, computes on the pieces, and then returns the results as a data frame. We use “,” to separate different items.

We only need item transactions, so remove customerID and Date columns.

```
itemList$CustomerID <- NULL
itemList$Date <- NULL
colnames(itemList) <- c(“items”)
```

Write the data fram to a csv file and check whether our transaction format is correct.

```
Write.csv(itemList,”market_basket.csv”,
 quote = FALSE, row.names = TRUE)
```

Perfect! Now we have our transaction dataset, and it shows the matrix of items being bought together. We don’t actually see how often they are bought together, and we don’t see rules either. But we are going to find out.

Let’s have a closer look at how many transactions we have and what they are.

```
Tr <- read.transactions(‘market_basket.csv’
, format = ‘basket’, sep=’,’)
Tr
Summary(tr)
```

The summary gives us some useful information:

- **Density:** The percentage of non-empty cells in the sparse matrix. In another words, the total number of items that are purchased divided by the total number of possible items in that matrix. We can calculate how many items were purchased using density like so: 19296 X 7881 X 0.0022
- The most frequent items should be the same as our results in Figure 3.
- Looking at the size of the transactions: 2247 transactions were for just 1 item, 1147 transactions for 2 items, all the way up to the biggest transaction: 1 transaction for 420 items. This indicates that most customers buy a small number of items in each transaction.
- The distribution of the data is right skewed.

```
itemFrequencyPlot(tr, topN=20,
type=’absolute’)
```

Create some rules

- We use the Apriori algorithm in Arules library to mine frequent itemsets and association rules. The algorithm employs level-wise search for frequent itemsets.
- We pass supp=0.001 and conf=0.8 to return all the rules that have a support of at least 0.1% and confidence of at least 80%.
- We sort the rules by decreasing confidence.
- Have a look at the summary of the rules.

```
Rules <- apriori(tr,
 parameter = list(supp=0.001,
 conf=0.8))
Rules <- sort(rules, by=’confidence’,
 decreasing = TRUE)
Summary(rules)
```

The summary of the rules gives us some very interesting information:

- The number of rules: 89,697.
- The distribution of rules by length: a length of 6 items has the most rules.
- The summary of quality measures: ranges of support, confidence, and lift.
- The information on data mining: total data mined, and the minimum parameters we set earlier.

```
Inspect(rules[1:10])
```

The interpretation is pretty straight forward:

- 100% customers who bought “WOBBLY CHICKEN” also bought “DECORATION”.
- 100% customers who bought “BLACK TEA” also bought “SUGAR JAR”.
- And plot these top 10 rules.

```
topRules <- rules[1:10]

plot(topRules)
```

Summary

In this post, we have learned how to perform Market Basket Analysis in R and how to interpret the results. If you want to implement them in Python, Mlxtend is a Python library that has an implementation of the Apriori algorithm for this sort of application.