Association Rule Extraction with R

1. Market Basket Dataset

The Basket dataset in the Data_Basket.csv file contains retail transactions (market baskets), with for each transaction information on the:

- Customer card number
- Payment method
- Socio-demographic data of the customer (gender, age, etc.)
- List of items purchased during the transaction

Dataset characteristics:

- Instances: 1000 each representing a transaction
- Number of variables: 18 discrete and continuous variables
- Missing values: None

The dataset variable name, type, description and domain of values are given in the table below.

Data Dictionary of the Basket Dataset

#	Variable name	Type	Description	Values
1	Card num.	Integer	Customer card number	[10150, 109884]
2	Amount	Real	Sales amount	[110.07, 498.86]
3	Payment	Text	Payment method	Cheque, Cash, Card
4	Gender	Boolean	Gender	M, F
5	Tenant	Boolean	Customer is tenant?	Yes, No
6	Income	Integer	Annual income in $	[102000, 300000]
7	Age	Integer	Age in whole years	[16, 50]
8	Fruits & vegetables	Boolean	Purchased item	Yes, No
9	Meat	Boolean	Purchased item	Yes, No
10	Milk products	Boolean	Purchased item	Yes, No
11	Canned vegetables	Boolean	Purchased item	Yes, No
12	Canned meat	Boolean	Purchased item	Yes, No
13	Frozen goods	Boolean	Purchased item	Yes, No
14	Beer	Boolean	Purchased item	Yes, No
15	Wine	Boolean	Purchased item	Yes, No
16	Soda drinks	Boolean	Purchased item	Yes, No
17	Fish	Boolean	Purchased item	Yes, No
18	Textile	Boolean	Purchased item	Yes, No

The purpose of this application is to discover links of association between:

- Purchased items,
- Customer gender and purchased items,
- Customer age group and purchased items.

We will use the *arules* and *aruleViz* libraries of the R software to extract knowledge patterns (itemsets and association rules) between the corresponding variables.

1.1. Loading the Data into R

☛ Download the Data_Basket.csv file from Jalon, and open it either with a spreadsheet (if necessary, define the column separator as the tab symbol and the decimal separator as the point to distinguish the columns) or a text editor (e.g., Notepad++) to verify the correct download of the dataset.

☛ Load the data from the Data_Basket.csv file into a `basket` data frame with the command:

```
> basket <- read.csv("Data_Basket.csv", header = TRUE, sep = ",", dec = ".")
```

☛ Identify in the help section of the `read.csv()` function what the `header = TRUE`, `sep = ","` and `dec = "."` parameters correspond to.

Note: In the case that the first line of the loaded file does not contain the names of the columns, R will name the variables V1, V2, V3, and so on.

☛ Display the content of the `basket` data frame in tabular format using the `View()` function:

```
> View(basket)
```

☛ Display the number of variables in the `basket` data frame using the `ncol()` function:

```
> ncol(basket)
```

☛ Display the names of the variables in the `basket` data frame using the `names()` function:

```
> names(basket)
```

☛ Display the summary statistics of the variables in the `basket` data frame using the `summary()` function.

☛ Attach the `basket` data frame using the `attach()` function so that its variables can be referenced by their name alone, i.e. without having to specify `basket$` before.

☛ Display the `Card.num.` variable using the `print()` function.

1.2. R Libraries for Extracting Frequent Patterns

We will use the *arules* and *arulesViz* libraries to extract and display association rules:

- The *arules* library provides functions for the extraction of frequent itemsets, maximal frequent itemsets, frequent closed itemsets, and association rules.
- The *arulesViz* library provides functions for graphical visualization of itemsets and association rules.

☛ Install/update the *arules* and *arulesViz* packages, and activate them into R.

☛ Display the help section of the `apriori()` function of the *arules* package by selecting in the lower-right zone of R Studio the tab named *Packages*, then clicking on the *arules* link and selecting the *apriori()* function in the *arules* documentation that appears.

2. Extracting Association Rules between Purchased Items

2.1. Data Preparation

For this extraction, only the variables corresponding to the items purchased (*Fruits & vegetables, Meat, Milk products, Canned vegetables, Canned meat, Frozen goods, Beer, Wine, Soda drinks, Fish and Textile*) will be used.

The variables *Card num.*, *Amount*, *Payment*, *Gender*, *Tenant*, *Income* and *Age* will be deleted using the selection operator `[,]` that can be used both to select and to delete (with negative parameters) a subset of rows and columns in a data frame.

☛ Delete the *Card num.* variable (first column) from the `basket` data frame with the selection operator `[,]` indicating by a negative value for the column parameter (2nd parameter) the deletion of the first column:

```
> basket <- basket [, -1]
```

☛ Delete the *Amount*, *Payment*, *Gender*, *Tenant*, *Income* and *Age* variables (six first columns now) with the command:

```
> basket <- basket [, -c(1:6)]
```

2.2. Association Rules Extraction Example

The extraction of association rules will be performed with the `apriori()` function of the *arules* library.

☛ Launch the extraction of association rules in the `basket` data frame for minimum support and confidence thresholds of 20% and 50% respectively by the command:

```
> rules1 <- apriori(basket, parameter = list (supp = 0.2, conf = 0.5, target =
  "rules", minlen = 2))
```

☛ Display summary information about the rule set generated by the command:

```
> summary(rules1)
```

Are displayed:

- The number of rules generated.
- The distribution of the rules according to their size (number of items).
- Summarized information on support, confidence and lift measures (quartiles and average).
- Execution information (dataset and minimum thresholds used).

☛ Display the first 20 rules generated by the command:

```
> inspect (rules1[1:20])
```

The extracted association rules are of the form:

```
     lhs              rhs                          support confidence lift      count
[1]  {Textile=Yes} => {Canned.vegetables=No}       0.205   0.7427536  1.0656436 205
```

in which:

- `lhs` is the antecedent,
- `rhs` is the consequence,
- `support` is the proportion of instances containing `lhs` and `rhs`,
- `confidence` is the proportion of instances containing `rhs` among those containing `lhs`,
- `lift` assesses the correlation between `lhs` and `rhs`: Lift(lhs → rhs) = P(lhs U rhs) / (P(lhs) x P(rhs)),
- `count` is the number of instances containing `lhs` and `rhs`.

☛ To simplify the display, set to 2 the number of decimal digits displayed using the `options()` function with the `digits` parameter, then re-display the first 20 rules generated:

```
> options(digits=2)
```

☛ Sort the `rules1` set of rules by decreasing confidence with the following command, then re-display the first 20 generated rules:

```
> rules1 <- sort(rules1, by = "confidence", decreasing = TRUE)
```

☛ Sort the `rules1` set by decreasing `lift` and identify how many rules have Lift > 1.

Extracted association rules contain items depicting both the presence (`Yes` value) and the absence (`No` value) of an item.

☛ In order to extract association rules between item presence only, replace the `"No"` values by the `NA` (Not Available) value in the `basket` data frame with the command:

```
> for (i in 1:ncol(basket)) basket[,i] <- replace(basket[,i], basket[,i]=="No",
  NA)
```

☛ Display the summary statistics of the modified variables in the `basket` data frame using the `summary()` function.

2.3. Top 10 Association Rules between Two Items

We want to extract the 10 most relevant association rules showing relationships between the purchases of two items: These rules must contain only one item in the antecedent and the consequent.

Use the `minlen` and `maxlen` parameters of the `apriori()` function to set respectively the minimal and maximal number of items in the extracted association rules.

If you get more than 10 rules with this form, select the rules with the highest value of lift (in case of identical lift, select those with the highest values for confidence, and then support).

☛ Extract the association rules using the `apriori()` function of the *arules* library according to the following process:

  ☞ The minimum support threshold will be varied between 20% and 10%, and the minimum confidence threshold between 50% and 20%.

  ☞ Perform successive tests first using the maximum values for the thresholds, then decreasing them successively until a satisfactory result is obtained.

  ☞ In order to ensure the statistical validity of the extracted rules, only those with Lift > 1 will be considered.

2.4. Graphical Representations of Association Rules

The `plot()` function of the *arulesViz* library allows to generate graphical representations of association rules.

☛ Display as a graph the extracted association rules by the command:

```
> plot(rule_set, method="graph")
```

☛ Display the extracted association rules with the `plot()` function using methods `"grouped"` and `"matrix"`.

By default, the color intensity depicts the Lift measure of the rule, dark colors representing the highest values. The `shading` parameter of the `plot()` function allows to depict the Confidence measure by color intensity.

☛ Display the extracted association rules with the `plot()` function using method `"matrix"` and color intensity to depict Confidence.

☛ Display as a graph the extracted association rules in interactive mode by the command:

```
> plot(rule_set, method="graph", shading="confidence", engine="interactive")
```

The menus of the window that appears allow you to modify and adjust the graphical representation.

☛ Display the extracted association rules with the `plot()` function using methods `"grouped"` and `"matrix"` in interactive mode.

Mouse commands allow to display information about the clicked item. Press Escape key to end the interactive display.

3. Extraction of Association Rules between Purchased Items and Other Variables

For the following exercises, the Data_Basket.csv file must be loaded into a new `basket` data frame using the `read.csv()` function.

In order to take into account this new data frame, the `detach(basket)` command must be executed and the `attach(basket)` command re-executed.

3.1. Distribution Histograms

We will use the functions of the *ggplot2* library in order to display the value distribution of variables in the `basket` data frame.

☛ In an Internet browser, go to the http://docs.ggplot2.org/ web-page that describes the many possibilities offered by *ggplot2*. If necessary, you will consult the description of the commands used in the following exercises to determine how to configure them.

☛ Install/update the *ggplot2* package and activate it into R.

We will use the `qplot(variable_name, data = dataset_name)` command to create simple univaried, i.e. one-dimensional, distribution diagrams.

☛ Display the distribution histogram of the `Gender` variable in the `basket` data frame by the command:

```
> qplot(Gender, data = basket)
```

You should see in the resulting histogram a bar for each value of `Gender`. This histogram allows us to verify that the two values, `Gender = M` and `Gender = F`, are "sufficiently" frequent in the data.

The `table(variable)` command generates a textual display of the distribution of the values of the variable received as a parameter.

☛ Display the count for each of the values of the `Gender` variable with the following command:

```
> table(Gender)
```

☛ Display the distribution histograms of the `Payment` and `Tenant` variables.

These histograms allow us to observe the distribution of the values within the value domain of each variable.

3.2. Comparison of Purchased Items by Gender

For this extraction, only the variables corresponding to the purchased items and the `Gender` variable will be used. The `Card num.`, `Amount`, `Payment`, `Tenant`, `Income` and `Age` variables will be ignored.

We want to determine which are the five most purchased items for each of the two values of the `Gender` variable.

Only association rules containing exclusively a value of the `Gender` variable, i.e. `Gender=M` or `Gender=F`, in the antecedent (`lhs`) and a single purchased item in the consequent (`rhs`) will be considered.

- ☛ See the help section of the `APappearance` parameter of the `apriori()` function. This parameter allows to define the content of the antecedent and/or the consequent of the extracted association rules.
- ☛ Extract association rules with `Gender=M` in the antecedent and one item in the consequent for minsupport and minconfidence thresholds of 20 % et 40 % respectively with the command:

```
> rules3a <- apriori(data = basket, parameter = list(supp=0.2, conf=0.4,
  minlen=2, maxlen=2), appearance = list(default="rhs", lhs="Gender=M"))
```

- ☛ Extract the association rules according to the following process:
  - ☞ The minimum support threshold will be varied between 20% and 5%, and the minimum confidence threshold between 40% and 10% until a satisfactory result is achieved.
  - ☞ Association rules with Lift ≤ 1 will be ignored.

3.3. Comparison of Purchased Items by Age

For this extraction, only the variables corresponding to the purchased items and the `Age` variable will be used. The variables `Card num.`, `Amount`, `Payment`, `Gender`, `Tenant` and `Income` will be deleted.

- ☛ Display the distribution histogram of the `Age` numeric variable in the `basket` data frame by the command:

```
> qplot(Age, data = basket)
```

Each bar corresponds to an interval of `Age` values, and the height of this bar indicates the number of corresponding instances in the `basket` data frame.

Setting the width of the histogram bars is possible with the `binwidth` parameter.

- ☛ Display the distribution histogram of the `Age` variable by setting the width of the histogram bars to 1 using the `binwidth` parameter to display a bar for each value of `Age`.

The distribution of the `Age` values is quite homogeneous over its value domain, and this variable will be discretized in 3 equal intervals (i.e. with the same amplitude min-max) using the `discretize()` function of the *arules* library.

To call the version of a function in a specific R package, several packages providing different implementations of the same function, you can use a command with the form `package::function()`. For example, the `discretize()` function of the *arules* package can be explicitly called with the command: `arules::discretize(…)`.

- ☛ Use the `arules::discretize()` function to discretize the `Age` variable in 3 equal intervals by the command:

```
> basket$Age <- arules::discretize(Age, method="interval", categories = 3)
```

- ☛ Use the `summary()` function to show the distribution counts of the discrete values (intervals) of the discretized `Age` variable.
- ☛ Use the `qplot()` function to display the distribution histogram of the discrete values of the discretized `Age` variable.

We want to determine which are the three most purchased items for each of the three `Age` value intervals.

Only association rules containing exclusively a value of the `Age` variable in antecedent and a single purchased item in the consequence will be considered.

- ☛ Extract the association rules according to the following process:
  - ☞ The minimum support threshold will be varied between 20% and 5%, and the minimum threshold of confidence between 50% and 20% until a satisfactory result is obtained.
  - ☞ Association rules with Lift ≤ 1 will be ignored.

3.4. Comparison of Purchased Items by Income

For this extraction, only the variables corresponding to the purchased items and the `Income` variable will be used. The variables `Card num.`, `Amount`, `Payment`, `Gender`, `Tenant` and `Age` will be deleted.

- ☛ Display the distribution histogram of the `Income` numeric variable in the `basket` data frame using the `qplot()` function.

We will reduce the number of bars in the histogram with the `bins` parameter.

- ☛ Display the distribution histogram of the `Income` variable, setting the number of bars of the histogram to 20 by the `bins` parameter, with the `qplot()` function.

The `Income` variable will be discretized into 4 intervals with identfal frequency (i.e. each interval corresponding to an identical, or nearly identical, number of instances) with the `arules::discretize()` function.

☛ Use the `arules::discretize()` function to discretize the `Income` variable into 4 intervals with identical frequency using the `method="frequency"` parameter.

☛ Use the `summary()` function to show the distribution counts of the discrete values (intervals) of the discretized `Income` variable.

☛ Use the `qplot()` function to display the distribution histogram of the discretized `Income` variable.

We want to determine which are the three most purchased items for each of the four `Income` value intervals obtained from the discretization.

Only association rules containing exclusively a value of `Income` in antecedent and a single purchased item in the consequence will be considered.

☛ Extract the association rules according to the following process:

☞ The minimum support threshold will be varied between 20% and 5%, and the minimum threshold of confidence between 50% and 20% until a satisfactory result is obtained.

☞ Association rules with Lift ≤ 1 will be ignored.

4. Subsidiary Exercises

The purpose of following exercises is to compare the frequent itemsets, maximal frequent itemsets and frequent closed itemsets extracted from the *basket* dataset.

For these exercises, the variables of the Data_Basket.csv dataset used will be:

- Purchased items: `Fruits & vegetables`, `Meat`, `Milk products`, `Canned vegetables`, `Canned meat`, `Frozen goods`, `Beer`, `Wine`, `Soda drinks`, `Fish` and `Textile` variables.
- Gender of the customer: `Gender` variable.
- Age of the customer: `Age` variable discretized in 3 equal-width intervals.
- Income of the customer: `Income` variable discretized in 4 equal-frequency intervals.

The four variables `Card num.`, `Amount`, `Payment` and `Tenant` will be disregarded.

☛ Extract the three types of itemsets from the `basket` data frame with the `apriori()` function for a minimum support threshold of 5% using:

☞ The `target = "frequent itemsets"` parameter for extracting frequent itemsets.

☞ The `target = "maximally frequent itemsets"` parameter for extracting maximal frequent itemsets.

☞ The `target = "closed frequent itemsets"` parameter for extracting frequent closed itemsets.

☛ Compare the number of itemsets in the three sets that can be obtained by applying the `length()` function to each set.

☛ Display the lists of itemsets in the three sets by applying the `inspect()` function to each set.

☛ Display a bar-plot for each of the three sets with the `barplot()` function. The example following command displays a bar-plot of the sizes of the itemsets in the `itemsets1` set:

```
> barplot(table(size(itemsets1)), xlab="Itemset size", ylab="Count")
```