Part 3 - Association Rules

Noah Kandie

Part 3: Association Rules

This section will require that you create association rules that will allow you to identify relationships between variables in the dataset. You are provided with a separate dataset that comprises groups of items that will be associated with others. Just like in the other sections, you will also be required to provide insights for your analysis. ## Load and Preview Dataset

```
# LOading arules library
library(arules)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'arules'
```

```
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
library(tinytex)
```

```
# load DAtaset
df3<-read.transactions('http://bit.ly/SupermarketDatasetII',sep = ",")
```

```
## Warning in asMethod(object): removing duplicated items in transactions
```

```
# Check class of dataset
class(df3)
```

```
## [1] "transactions"
## attr(,"package")
## [1] "arules"
```

```
# First 5 transcations
inspect(df3[1:5])
```

```
##     items
## [1] {almonds,
##      antioxydant juice,
##      avocado,
```

```
##      cottage cheese,
##      energy drink,
##      frozen smoothie,
##      green grapes,
##      green tea,
##      honey,
##      low fat yogurt,
##      mineral water,
##      olive oil,
##      salad,
##      salmon,
##      shrimp,
##      spinach,
##      tomato juice,
##      vegetables mix,
##      whole weat flour,
##      yams}
## [2] {burgers,
##      eggs,
##      meatballs}
## [3] {chutney}
## [4] {avocado,
##      turkey}
## [5] {energy bar,
##      green tea,
##      milk,
##      mineral water,
##      whole wheat rice}
```

```
#summary
summary(df3)
```

```
## transactions as itemMatrix in sparse format with
##  7501 rows (elements/itemsets/transactions) and
##  119 columns (items) and a density of 0.03288973
##
## most frequent items:
## mineral water          eggs     spaghetti  french fries     chocolate
##          1788          1348          1306          1282          1229
##       (Other)
##         22405
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
## 1754 1358 1044  816  667  493  391  324  259  139  102   67   40   22   17    4
##   18   19   20
##    1    2    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   3.914   5.000  20.000
##
## includes extended item information - examples:
##              labels
```

```
## 1           almonds
## 2 antioxydant juice
## 3         asparagus
```

most frequent items: -mineral water -eggs -spaghetti
-french fries -chocolate

```
# Exploring the frequency of some articles
# i.e. transacations ranging from 8 to 10 and performing
# some operation in percentage terms of the total transactions
#
itemFrequency(df3[, 5:10],type = "absolute")
```

```
##    babies food          bacon barbecue sauce      black tea    blueberries
##             34             65              81            107             69
##     body spray
##             86
```

```
round(itemFrequency(df3[, 5:10],type = "relative")*100,2)
```

```
##    babies food          bacon barbecue sauce      black tea    blueberries
##           0.45           0.87            1.08           1.43           0.92
##     body spray
##           1.15
```

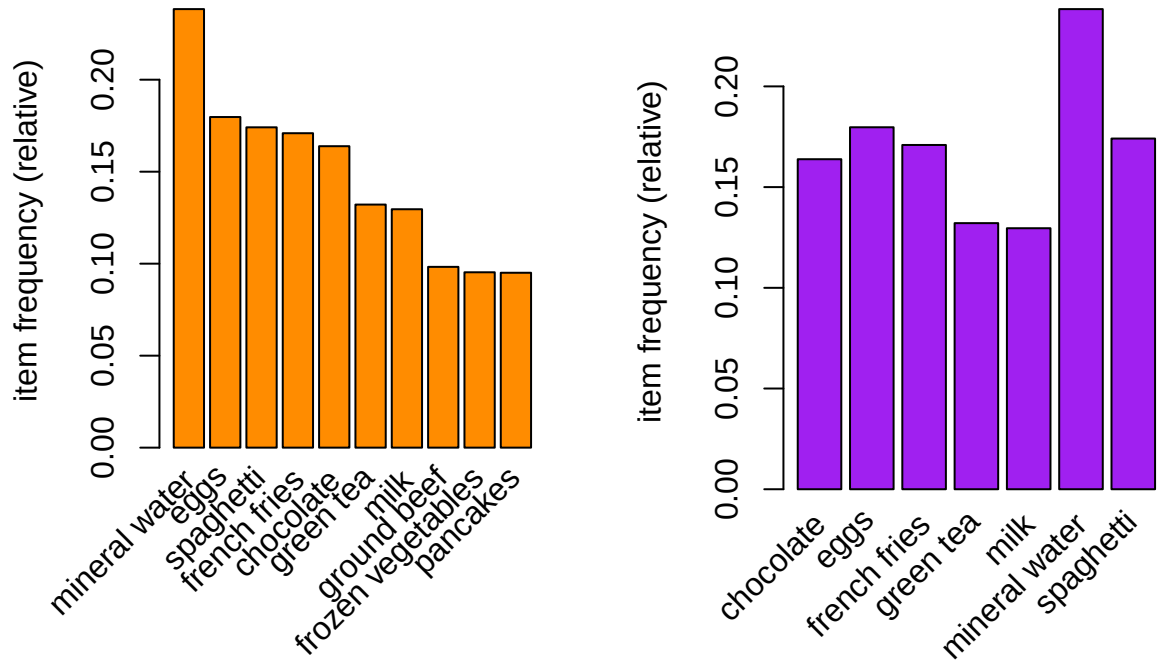
```
# preview of  the items that make up our dataset,

items<-as.data.frame(itemLabels(df3))
colnames(items) <- "Item"
head(items, 20)
```

```
##                Item
## 1           almonds
## 2  antioxydant juice
## 3          asparagus
## 4           avocado
## 5        babies food
## 6              bacon
## 7     barbecue sauce
## 8          black tea
## 9        blueberries
## 10        body spray
## 11           bramble
## 12          brownies
## 13         bug spray
## 14      burger sauce
## 15           burgers
## 16            butter
## 17              cake
## 18        candy bars
## 19           carrots
## 20       cauliflower
```

```
# Displaying top 10 most common items in the transactions dataset
# and the items whose relative importance is at least 10%
#
par(mfrow = c(1, 2))

# plot the frequency of items
itemFrequencyPlot(df3, topN = 10,col='darkorange')
itemFrequencyPlot(df3, support = 0.1,col="purple")
```




```
# Building a model based on association rules using the apriori function
# We use Min Support as 0.001 and confidence as 0.8

rules <- apriori (df3, parameter = list(supp = 0.001, conf = 0.8))
```

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

```
##
## Absolute minimum support count: 7
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[119 item(s), 7501 transaction(s)] done [0.02s].
## sorting and recoding items ... [116 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 done [0.01s].
## writing ... [74 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rules
```

```
## set of 74 rules
```

```
#Summary of the model
summary(rules)
```

```
## set of 74 rules
##
## rule length distribution (lhs + rhs):sizes
##  3  4  5  6
## 15 42 16  1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   4.000   4.000   4.041   4.000   6.000
##
## summary of quality measures:
##     support           confidence        coverage             lift
##  Min.   :0.001067   Min.   :0.8000   Min.   :0.001067   Min.   : 3.356
##  1st Qu.:0.001067   1st Qu.:0.8000   1st Qu.:0.001333   1st Qu.: 3.432
##  Median :0.001133   Median :0.8333   Median :0.001333   Median : 3.795
##  Mean   :0.001256   Mean   :0.8504   Mean   :0.001479   Mean   : 4.823
##  3rd Qu.:0.001333   3rd Qu.:0.8889   3rd Qu.:0.001600   3rd Qu.: 4.877
##  Max.   :0.002533   Max.   :1.0000   Max.   :0.002666   Max.   :12.722
##      count
##  Min.   : 8.000
##  1st Qu.: 8.000
##  Median : 8.500
##  Mean   : 9.419
##  3rd Qu.:10.000
##  Max.   :19.000
##
## mining info:
##  data ntransactions support confidence
##   df3          7501   0.001        0.8
```

Cross Validation

```
# Building a apriori model with Min Support as 0.002 and confidence as 0.8.
rules2 <- apriori (df3,parameter = list(supp = 0.002, conf = 0.8))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5   0.002      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 15
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[119 item(s), 7501 transaction(s)] done [0.04s].
## sorting and recoding items ... [115 item(s)] done [0.00s].
## creating transaction tree ... done [0.02s].
## checking subsets of size 1 2 3 4 5 done [0.02s].
## writing ... [2 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
# Building apriori model with Min Support as 0.002 and confidence as 0.6.
rules3 <- apriori (df3, parameter = list(supp = 0.001, conf = 0.6))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5   0.001      1
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 7
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[119 item(s), 7501 transaction(s)] done [0.01s].
## sorting and recoding items ... [116 item(s)] done [0.00s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 done [0.03s].
## writing ... [545 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
rules
```

```
## set of 74 rules
```

```
rules2
```

```
## set of 2 rules
```

```
rules3
```

```
## set of 545 rules
```

From this we get the best parametered model is rule 1. It has a good size of rules

Inspection

```
#We now inspect the model
rules<-sort(rules, by="confidence", decreasing=TRUE)
inspect(rules[1:5])
```

```
##     lhs                                          rhs             support
## [1] {french fries,mushroom cream sauce,pasta} => {escalope}      0.001066524
## [2] {ground beef,light cream,olive oil}       => {mineral water} 0.001199840
## [3] {cake,meatballs,mineral water}            => {milk}          0.001066524
## [4] {cake,olive oil,shrimp}                   => {mineral water} 0.001199840
## [5] {mushroom cream sauce,pasta}              => {escalope}      0.002532996
##     confidence coverage    lift      count
## [1] 1.00       0.001066524 12.606723  8
## [2] 1.00       0.001199840  4.195190  9
## [3] 1.00       0.001066524  7.717078  8
## [4] 1.00       0.001199840  4.195190  9
## [5] 0.95       0.002666311 11.976387 19
```

We see that mineral water and escalope are the most bought product with a high confidence of <95%. So we will check them

Escalope

```
# we could create a subset of rules concerning these products
escalope <- subset(rules, subset = rhs %pin% "escalope")

# Then order by confidence
escalope<-sort(escalope, by="confidence", decreasing=TRUE)
inspect(escalope[1:2])
```

```
##     lhs                                          rhs        support
## [1] {french fries,mushroom cream sauce,pasta} => {escalope} 0.001066524
## [2] {mushroom cream sauce,pasta}              => {escalope} 0.002532996
##     confidence coverage    lift     count
## [1] 1.00       0.001066524 12.60672  8
## [2] 0.95       0.002666311 11.97639 19
```

Seems escalope has two transactions. The product bought together are mushroom cream sauce and pasta

Mineral Water

```
# Subset the rules
mineral_water <- subset(rules, subset = rhs %pin% "mineral water")

# Order by confidence
yogurt<-sort(mineral_water, by="confidence", decreasing=TRUE)

# inspect top 5
inspect(mineral_water[1:5])
```

```
##     lhs                    rhs                support confidence    coverage     lift count
## [1] {ground beef,
##      light cream,
##      olive oil}         => {mineral water} 0.001199840  1.0000000 0.001199840 4.195190     9
## [2] {cake,
##      olive oil,
##      shrimp}            => {mineral water} 0.001199840  1.0000000 0.001199840 4.195190     9
## [3] {red wine,
##      soup}              => {mineral water} 0.001866418  0.9333333 0.001999733 3.915511    14
## [4] {ground beef,
##      pancakes,
##      whole wheat rice}  => {mineral water} 0.001333156  0.9090909 0.001466471 3.813809    10
## [5] {frozen vegetables,
##      milk,
##      spaghetti,
##      turkey}            => {mineral water} 0.001199840  0.9000000 0.001333156 3.775671     9
```

##Observations - Escalope to be shelved closer to mushroom cream sauce and pasta with a confidence of 1 - Mineral water are associated with the following products: Olive oil and ground beef, with pasta(family) are also linked

Who previously bought product

```
# Escalope
# Subset the rules
escalope <- subset(rules, subset = lhs %pin% "escalope")

# Order by confidence
escalope<-sort(escalope, by="confidence", decreasing=TRUE)

# inspect top 5
inspect(escalope[1:2])
```

```
##     lhs                                rhs         support     confidence
## [1] {escalope,hot dogs,mineral water} => {milk}      0.001066524 0.8888889
## [2] {escalope,french fries,shrimp}    => {chocolate} 0.001066524 0.8888889
##     coverage   lift     count
## [1] 0.00119984 6.859625 8
## [2] 0.00119984 5.425188 8
```

```r
# Escalope
# Subset the rules
mineral_water <- subset(rules, subset = lhs %pin% "mineral water")

# Order by confidence
mineral_water<-sort(mineral_water, by="confidence", decreasing=TRUE)

# inspect top 5
inspect(mineral_water[1:5])
```

```
##     lhs                                    rhs           support     confidence
## [1] {cake,meatballs,mineral water}     => {milk}        0.001066524 1.0000000
## [2] {eggs,mineral water,pasta}         => {shrimp}      0.001333156 0.9090909
## [3] {herb & pepper,mineral water,rice} => {ground beef} 0.001333156 0.9090909
## [4] {light cream,mineral water,shrimp} => {spaghetti}   0.001066524 0.8888889
## [5] {grated cheese,mineral water,rice} => {ground beef} 0.001066524 0.8888889
##     coverage    lift      count
## [1] 0.001066524  7.717078  8
## [2] 0.001466471 12.722185 10
## [3] 0.001466471  9.252498 10
## [4] 0.001199840  5.105326  8
## [5] 0.001199840  9.046887  8
```

Observation

i) milk and chocolate were previously bought with escalope at a 89% confidence
ii) milk was regularly alongside mineral water with confidence of 91%