Market Basket Analysis: groceries.csv

Isha Bandi

17 June 2017

Market Basket Analysis

Uses public data: groceries Importing all the required libraries.

```
library(reshape2) #for 'melting' the data into tidier long format
library(data.table) #for playing with data faster
library(arules)
```

Importing dataset as data.table and adding the column names.

```
g = data.table(read.table('E:\\Isha\\mbadataset\\groceries.csv',sep = ',',na.strings =
c('NA',''),dec = '.',header=F,fill=TRUE,stringsAsFactors = T))
colnames(g) = c('p1','p2','p3','p4')
head(g)
```

```
##                  p1                  p2             p3
## 1:     citrus fruit semi-finished bread      margarine
## 2:   tropical fruit              yogurt         coffee
## 3:       whole milk                  NA             NA
## 4:        pip fruit              yogurt  cream cheese
## 5: other vegetables          whole milk condensed milk
## 6:       whole milk              butter         yogurt
##                          p4
## 1:              ready soups
## 2:                       NA
## 3:                       NA
## 4:             meat spreads
## 5: long life bakery product
## 6:                     rice
```

The dataset is already in the basket format transactions i.e. each row represents one transaction. Hence, we can generate rules using apriori from arules package. We can adjust the minimum support value and minimum confidence. We can also specify the max number of items in one set using maxlen. I have kept small parametres to generate more number of rules.

```
rules1 = apriori(g,parameter = list(supp=0.001,conf=0.08))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.08    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: 15
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[652 item(s), 15296 transaction(s)] done [0.02s].
## sorting and recoding items ... [419 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.01s].
## writing ... [646 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

Summary of the rules generated.

```
summary(rules1)
```

```
## set of 646 rules
##
## rule length distribution (lhs + rhs):sizes
##   2   3
## 527 119
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   2.000   2.184   2.000   3.000
##
## summary of quality measures:
##     support           confidence          lift
##  Min.   :0.001046   Min.   :0.0800   Min.   :  1.403
##  1st Qu.:0.001242   1st Qu.:0.1111   1st Qu.:  3.961
##  Median :0.001634   Median :0.1553   Median :  7.282
##  Mean   :0.002230   Mean   :0.2241   Mean   : 10.937
##  3rd Qu.:0.002729   3rd Qu.:0.2679   3rd Qu.: 12.323
##  Max.   :0.014514   Max.   :1.0000   Max.   :132.433
##
## mining info:
##  data ntransactions support confidence
##     g         15296   0.001       0.08
```

Sorting rules according to the confidence

```
rules1=sort(rules1,by='confidence')
inspect(rules1[1:4])
```

```
##     lhs                   rhs                      support confidence      lift
## [1] {p2=sausage}       => {p1=frankfurter}      0.006472280          1  26.37241
## [2] {p2=citrus fruit,
##      p4=pip fruit}     => {p3=tropical fruit}   0.001046025          1 106.96503
## [3] {p2=onions,
##      p4=whole milk}    => {p3=other vegetables} 0.001111402          1  34.68481
## [4] {p1=citrus fruit,
##      p3=pip fruit}     => {p2=tropical fruit}   0.002026674          1  42.60724
```

Removing redendancies from the rules.

```
rules1 = rules1[!is.redundant(rules1)]
rules1
```

```
## set of 617 rules
```

Now, targeting products ‘whole milk’ i.e. generating rules to see what products were bought when ‘whole milk’ was bought as p2.

```
class(g$p1)
```

```
## [1] "factor"
```

```
rulesforwm = apriori(g,parameter = list(supp=0.001,conf=0.08
),appearance=list(default='rhs',lhs='p2=whole milk'))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.08    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: 15
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[652 item(s), 15296 transaction(s)] done [0.02s].
## sorting and recoding items ... [419 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 done [0.00s].
## writing ... [6 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(rulesforwm)
```

```
##     lhs                rhs                    support     confidence
## [1] {p2=whole milk} => {p3=curd}             0.004053347 0.08344549
## [2] {p2=whole milk} => {p3=butter}           0.004641736 0.09555855
## [3] {p2=whole milk} => {p3=yogurt}           0.007191423 0.14804845
## [4] {p2=whole milk} => {p3=rolls/buns}       0.003987971 0.08209960
## [5] {p2=whole milk} => {p1=tropical fruit}   0.004968619 0.10228802
## [6] {p2=whole milk} => {p1=other vegetables} 0.013663703 0.28129206
##     lift
## [1] 10.816799
## [2] 10.440454
## [3]  6.526078
## [4]  3.163213
## [5]  3.167202
## [6]  7.380177
```