Performing Market Basket Analysis using Association Rules

Saurabh Yelne

7/26/2018

Loading the arules library and the data set into sparse matrix format

Please find the rpubs link for this analysis Market_Basket_Analysis

```
suppressPackageStartupMessages(library(arules))
```

```
## Warning: package 'arules' was built under R version 3.4.4
```

```
groceries <- read.transactions("http://www.sci.csueastbay.edu/~esuess/classes/Statistics_6620/Presentati
                               sep = ",")
```

Exploring groceries dataset

```
summary(groceries)
```

```
## transactions as itemMatrix in sparse format with
##  9835 rows (elements/itemsets/transactions) and
##  169 columns (items) and a density of 0.02609146
##
## most frequent items:
##       whole milk other vegetables       rolls/buns             soda
##             2513             1903             1809             1715
##           yogurt          (Other)
##             1372            34055
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 2159 1643 1299 1005  855  645  545  438  350  246  182  117   78   77   55
##   16   17   18   19   20   21   22   23   24   26   27   28   29   32
##   46   29   14   14    9   11    4    6    1    1    1    1    3    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   3.000   4.409   6.000  32.000
##
## includes extended item information - examples:
##             labels
## 1 abrasive cleaner
## 2 artif. sweetener
## 3   baby cosmetics
```

There are 9835 transactions and 169 items. Density is 2.6% which means there are 2.6% nonzero matrix cells which are 9835x169 = 1662115, 1662115x0.02609146 = 43,367. Whole milk is the most frequent item.

Further we get statistics about the size of the transactions. We can see 2159 transaction has 1 item and so on. The average items/transaction are 4.409.

Looking at the first 5 transactions and the proportion of transaction that contains items

```r
inspect(groceries[1:5])
```

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

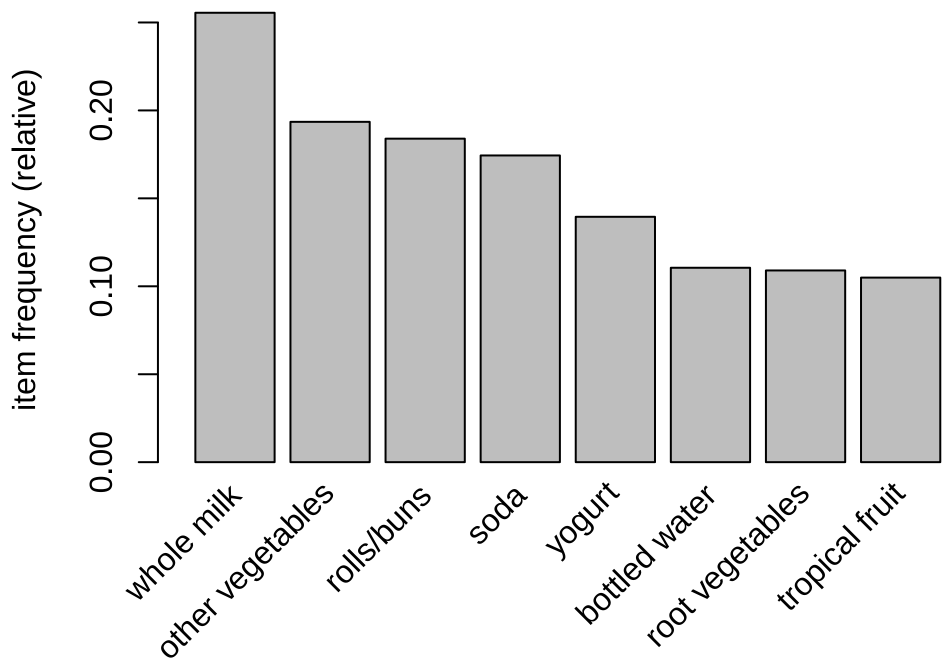
```r
itemFrequency(groceries[,1:5])
```

```
## abrasive cleaner artif. sweetener   baby cosmetics        baby food
##     0.0035587189     0.0032536858     0.0006100661     0.0001016777
##             bags
##     0.0004067107
```

We can see abrasive cleaner is present in 3.55% of transactions while artif. sweetener in 3.25% and so on. The items are displayed according to alphabetical order.

Visualizating Item Frequency Plot with top 10 most frequent item

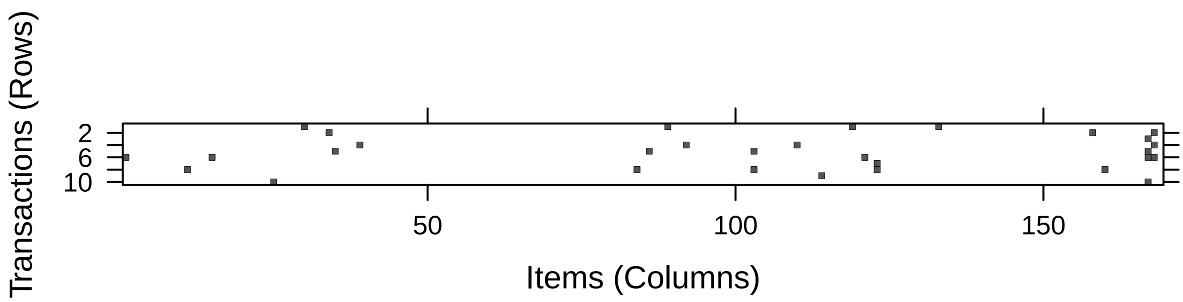
```r
itemFrequencyPlot(groceries,support=0.1, topN=10)
```



The histogram shows the eight items in the groceries data with at least 10 percent support.

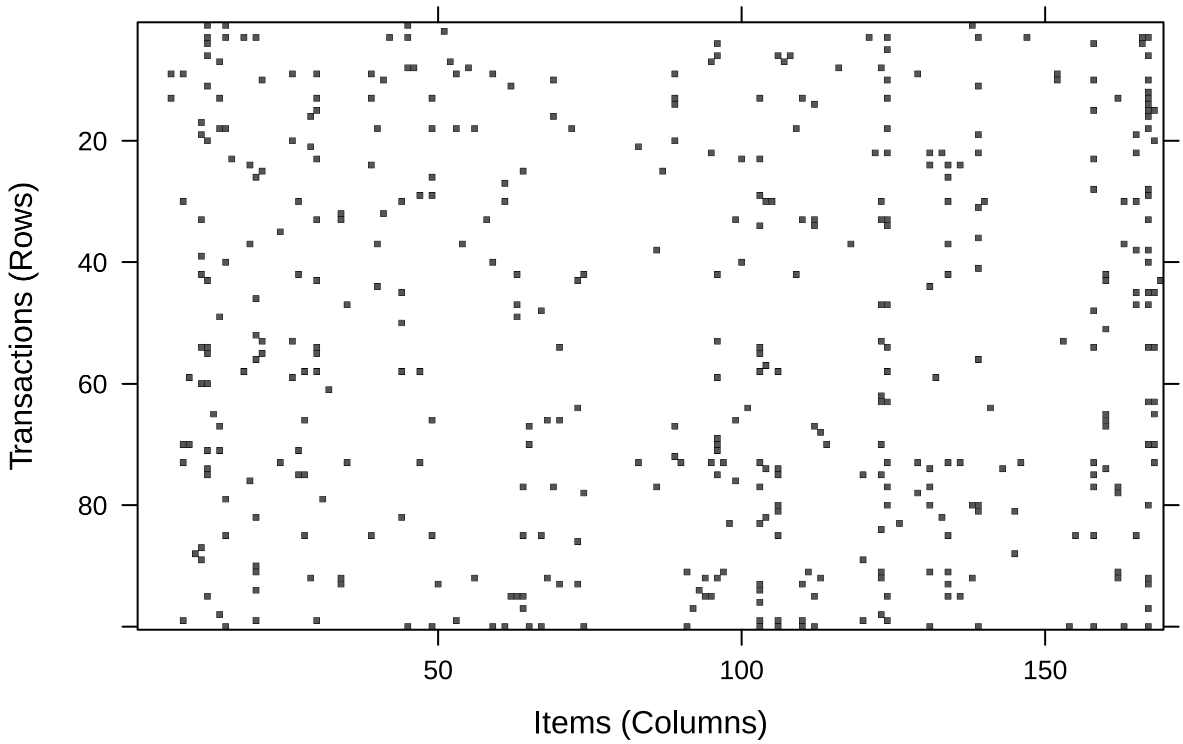
Visualizing the sparse matrix

```
image(groceries[1:10])
```



The **image** function displays first 10 rows and 169 columns. Cells in the matrix are filled with black for transactions (rows) where the item (column) was purchased.

```
image(sample(groceries, 100))
```



Using **sample** function with image function creates a big visualization of the sparse matrix. A random 100 sample is plotted and thus we can get insights about the items in a transaction. We can see that some columns are heavily populated with the black dots indicating those items are more popular and are present in many transactions. Lets continue with our analysis further.

Training a model with Apriori Algorithm

With the transaction data we have, we can find the association between the items in the dataset using Apriori algorithm from the arules package. For finding the association rules, support and confidence parameter plays a vital role. If these are set high then there will be few or no rules and setting it low can give very high number of unreliable rules and the operation will take a lot of time and may run out of memory. Lets say we are interested in finding out items which are sold twice a day, 60 times a month which equals 60/ 9835 = 0.006. This is our setting for **support** parameter. We can keep **confidence** level to 0.25. **minlen** is set to 2 to remove rules that contain less than two items.

```
asso_rules <- apriori(groceries, parameter = list(support =
                            0.006, confidence = 0.25, minlen = 2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.25    0.1    1 none FALSE            TRUE       5   0.006      2
##  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: 59
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [109 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [463 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

Evaluating the model

```r
summary(asso_rules)
```

```
## set of 463 rules
##
## rule length distribution (lhs + rhs):sizes
##   2   3   4
## 150 297  16
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   3.000   2.711   3.000   4.000
##
## summary of quality measures:
##     support           confidence          lift            count
##  Min.   :0.006101   Min.   :0.2500   Min.   :0.9932   Min.   : 60.0
##  1st Qu.:0.007117   1st Qu.:0.2971   1st Qu.:1.6229   1st Qu.: 70.0
##  Median :0.008744   Median :0.3554   Median :1.9332   Median : 86.0
##  Mean   :0.011539   Mean   :0.3786   Mean   :2.0351   Mean   :113.5
##  3rd Qu.:0.012303   3rd Qu.:0.4495   3rd Qu.:2.3565   3rd Qu.:121.0
##  Max.   :0.074835   Max.   :0.6600   Max.   :3.9565   Max.   :736.0
##
## mining info:
##       data ntransactions support confidence
##  groceries          9835   0.006       0.25
```

By using **summary** function we can get a detailed overview of the association rules. As we can see there are 463 association rules. Rule length distribution tells us how many items are present in how many rules. 2 items are present in 150 rules, 3 in 297 rules and 4 in 16 rules.

To further our analysis we can inspect individual rules by using **inspect** function.

```r
inspect(asso_rules[1:5])
```

```
##     lhs                 rhs                support     confidence lift
## [1] {potted plants} => {whole milk}       0.006914082 0.4000000  1.565460
## [2] {pasta}         => {whole milk}       0.006100661 0.4054054  1.586614
## [3] {herbs}         => {root vegetables}  0.007015760 0.4312500  3.956477
## [4] {herbs}         => {other vegetables} 0.007727504 0.4750000  2.454874
## [5] {herbs}         => {whole milk}       0.007727504 0.4750000  1.858983
##     count
## [1] 68
## [2] 60
## [3] 69
## [4] 76
```

```
## [5] 76
```

```r
itemsets<-eclat(groceries)[1:5]
```

```
## Eclat
##
## parameter specification:
##  tidLists support minlen maxlen            target   ext
##     FALSE     0.1      1     10 frequent itemsets FALSE
##
## algorithmic control:
##  sparse sort verbose
##       7   -2    TRUE
##
## Absolute minimum support count: 983
##
## create itemset ...
## set transactions ...[169 item(s), 9835 transaction(s)] done [0.00s].
## sorting and recoding items ... [8 item(s)] done [0.00s].
## creating bit matrix ... [8 row(s), 9835 column(s)] done [0.00s].
## writing  ... [8 set(s)] done [0.00s].
## Creating S4 object  ... done [0.00s].
```

```r
support(items(itemsets), groceries)
```

```
## [1] 0.2555160 0.1934926 0.1839349 0.1395018 0.1743772
```

The first five rules are seen here. Also, we can see support for the top 5 most frequent items. We can see the **lift** column along with support and confidence. *The lift of a rule measures how much likely an item or itemset is purchased relative to its typical rate of purchase, given that you know another item or itemsethas been purchased.*

Sorting the association rules according to lift

```r
inspect(sort(asso_rules, by = "lift")[1:5])
```

```
##     lhs                     rhs                      support confidence     lift count
## [1] {herbs}              => {root vegetables}    0.007015760  0.4312500 3.956477    69
## [2] {berries}            => {whipped/sour cream} 0.009049314  0.2721713 3.796886    89
## [3] {other vegetables,
##      tropical fruit,
##      whole milk}         => {root vegetables}    0.007015760  0.4107143 3.768074    69
## [4] {beef,
##      other vegetables} => {root vegetables}    0.007930859  0.4020619 3.688692    78
## [5] {other vegetables,
##      tropical fruit}   => {pip fruit}          0.009456024  0.2634561 3.482649    93
```

We can sort the rules according to support, confidence or lift. Here the first rule is with the highest lift which indicates that customers who buy herbs are almost 4 times more likely to buy root vegetables verses other customers.

Taking subsets of association rules

Sometimes the marketing team requires to promote a specific product, say they want to promote berries, and want to find out how often and with which items the berries are purchased. The **subset** function enables one to find subsets of transactions, items or rules. The **%in%** operator is used for exact matching

```
berryrules <- subset(asso_rules, items %in% "berries")
inspect(berryrules)
```

```
##     lhs          rhs                  support     confidence lift
## [1] {berries} => {whipped/sour cream} 0.009049314 0.2721713  3.796886
## [2] {berries} => {yogurt}             0.010574479 0.3180428  2.279848
## [3] {berries} => {other vegetables}   0.010269446 0.3088685  1.596280
## [4] {berries} => {whole milk}         0.011794611 0.3547401  1.388328
##     count
## [1]  89
## [2] 104
## [3] 101
## [4] 116
```

We find 4 rules related to berries 2 of which seems interesting and gives useful insight. In addition to whipped cream, berries are also purchased with yogurt frequently that could serve well for breakfast, lunch and dessert.

```
vegrules <- subset(asso_rules, items %pin% "vegetable")
inspect(sort(vegrules, by="lift")[1:10])
```

```
##      lhs                     rhs                    support confidence     lift count
## [1]  {herbs}              => {root vegetables} 0.007015760  0.4312500 3.956477    69
## [2]  {other vegetables,
##       tropical fruit,
##       whole milk}         => {root vegetables} 0.007015760  0.4107143 3.768074    69
## [3]  {beef,
##       other vegetables} => {root vegetables} 0.007930859  0.4020619 3.688692    78
## [4]  {other vegetables,
##       tropical fruit}   => {pip fruit}       0.009456024  0.2634561 3.482649    93
## [5]  {beef,
##       whole milk}       => {root vegetables} 0.008032537  0.3779904 3.467851    79
## [6]  {other vegetables,
##       pip fruit}        => {tropical fruit}  0.009456024  0.3618677 3.448613    93
## [7]  {citrus fruit,
##       other vegetables} => {root vegetables} 0.010371124  0.3591549 3.295045   102
## [8]  {other vegetables,
##       whole milk,
##       yogurt}           => {tropical fruit}  0.007625826  0.3424658 3.263712    75
## [9]  {other vegetables,
##       whole milk,
##       yogurt}           => {root vegetables} 0.007829181  0.3515982 3.225716    77
## [10] {other vegetables,
##       tropical fruit,
##       whole milk}       => {yogurt}          0.007625826  0.4464286 3.200164    75
```

Now we can see top 10 rules with highest lift which has some kind of vegetable in the transaction. The **%pin%** operator is used to partial matching.We can see herbs and root vegetables are purchased four times more often together as seen before. Also it is very useful to know that customers buying other vegetables,tropical fruit and whole milk are 3.77 times more likely to buy root vegetables. Thus we can find several other useful rules and our marketing team can plan their promotional offers likewise.

THANK YOU SO MUCH FOR READING