M4L4\_Assignment

Xueyi Fan

June 13, 2016

# Assignment:

* First, select a transaction dataset from the Frequent Itemset Mining Dataset Repository at <http://fimi.ua.ac.be/data/> or another transaction dataset of your choice from the Web.
* Next, generate a set of 50 or so (non-redundant) rules.
* Finally answer the following questions:

1. Which rules make sense to you? Highlight the five best and five worst of your rule set.
2. How did you choose the level of support and confidence?
3. What is the lift and conviction of your best and worst rules?
4. Visualize your 50 association rules. Where do the best and worst end up in your plot?
5. Does the model make sense?

I choose the [retail.dat dataset](http://fimi.ua.ac.be/data/) from Frequent Itemset Mining Data Repository which contains the (anonymized) retail market basket data from an anonymous Belgian retail store.

Loading the data:

require("arules")

## Loading required package: arules

## Warning: package 'arules' was built under R version 3.2.5

## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 3.2.5

##   
## Attaching package: 'arules'

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

require("arulesViz")

## Loading required package: arulesViz

## Warning: package 'arulesViz' was built under R version 3.2.5

## Loading required package: grid

data\_url <- 'http://fimi.ua.ac.be/data/retail.dat'  
mydata<- read.transactions(url(data\_url))  
  
summary(mydata)

## transactions as itemMatrix in sparse format with  
## 88162 rows (elements/itemsets/transactions) and  
## 16470 columns (items) and a density of 0.0006257289   
##   
## most frequent items:  
## 39 48 38 32 41 (Other)   
## 50675 42135 15596 15167 14945 770058   
##   
## element (itemset/transaction) length distribution:  
## sizes  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15   
## 3016 5516 6919 7210 6814 6163 5746 5143 4660 4086 3751 3285 2866 2620 2310   
## 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30   
## 2115 1874 1645 1469 1290 1205 981 887 819 684 586 582 472 480 355   
## 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45   
## 310 303 272 234 194 136 153 123 115 112 76 66 71 60 50   
## 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60   
## 44 37 37 33 22 24 21 21 10 11 10 9 11 4 9   
## 61 62 63 64 65 66 67 68 71 73 74 76   
## 7 4 5 2 2 5 3 3 1 1 1 1   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.00 4.00 8.00 10.31 14.00 76.00   
##   
## includes extended item information - examples:  
## labels  
## 1 0  
## 2 1  
## 3 10

inspect(mydata[1:5])

## items  
## 1 {0,   
## 1,   
## 10,   
## 11,   
## 12,   
## 13,   
## 14,   
## 15,   
## 16,   
## 17,   
## 18,   
## 19,   
## 2,   
## 20,   
## 21,   
## 22,   
## 23,   
## 24,   
## 25,   
## 26,   
## 27,   
## 28,   
## 29,   
## 3,   
## 4,   
## 5,   
## 6,   
## 7,   
## 8,   
## 9}   
## 2 {30,   
## 31,   
## 32}   
## 3 {33,   
## 34,   
## 35}   
## 4 {36,   
## 37,   
## 38,   
## 39,   
## 40,   
## 41,   
## 42,   
## 43,   
## 44,   
## 45,   
## 46}   
## 5 {38,   
## 39,   
## 47,   
## 48}

### generate a set of 50 or so (non-redundant) rules.

rules <- apriori(mydata, parameter = list(support = 0.01, confidence = 0.60, target = "rules"))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport support minlen maxlen  
## 0.6 0.1 1 none FALSE TRUE 0.01 1 10  
## target ext  
## rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 881   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[16470 item(s), 88162 transaction(s)] done [0.15s].  
## sorting and recoding items ... [70 item(s)] done [0.01s].  
## creating transaction tree ... done [0.04s].  
## checking subsets of size 1 2 3 4 done [0.01s].  
## writing ... [84 rule(s)] done [0.00s].  
## creating S4 object ... done [0.02s].

rules

## set of 84 rules

rules.50 <- rules[1:50]  
inspect(rules.50[1:3])

## lhs rhs support confidence lift   
## 1 {37} => {38} 0.01186452 0.9739292 5.505485  
## 2 {286} => {38} 0.01265852 0.9433643 5.332706  
## 3 {12925} => {39} 0.01063950 0.6394001 1.112399

#remove the redundant rules  
rules.50.sorted = sort(rules.50, by="lift")  
subset.matrix <- is.subset(rules.50.sorted, rules.50.sorted)

## Warning: closing unused connection 5 (http://fimi.ua.ac.be/data/retail.dat)

subset.matrix[lower.tri(subset.matrix, diag=T)] <- NA   
redundant <- colSums(subset.matrix, na.rm=T) >=1  
which(redundant)

## {39,475,48} {36,38,39} {170,38,39} {110,38,39}   
## 18 28 37 44

rules.50.pruned <- rules.50.sorted[!redundant]  
rules.50.pruned

## set of 46 rules

### 1. Which rules make sense to you? Highlight the five best and five worst of your rule set.

inspect(rules.50.pruned[1:5])

## lhs rhs support confidence lift   
## 40 {110,39} => {38} 0.01973639 0.9891984 5.591800  
## 48 {170,48} => {38} 0.01744516 0.9877970 5.583878  
## 38 {110,48} => {38} 0.01543749 0.9862319 5.575030  
## 50 {170,39} => {38} 0.02290102 0.9805731 5.543042  
## 24 {170} => {38} 0.03437989 0.9780574 5.528821

inspect(rules.50.sorted[42:46])

## lhs rhs support confidence lift   
## 3 {12925} => {39} 0.01063950 0.6394001 1.112399  
## 16 {147} => {39} 0.01289671 0.6391231 1.111917  
## 39 {110,38} => {39} 0.01973639 0.6385321 1.110888  
## 29 {237} => {39} 0.02188018 0.6362137 1.106855  
## 15 {110} => {39} 0.01995191 0.6295634 1.095285

According to the result the rule {110,39} => {38} which means the person who but the item 110 and 39 will but item 38. Because this rule have a high confidence which is 98.9%.

The five best rule set are:  
{110,39} => {38}  
{170,48} => {38}  
{110,48} => {38}  
{170,39} => {38}  
{170} => {38}

The five worst rule set are: {12925} => {39}  
{147} => {39}  
{110,38} => {39}  
{237} => {39}  
{110} => {39}

### 2. How did you choose the level of support and confidence?

If I use default parameters, it will not generate any rules, because the default parameters of apriori function are support = 0.1, confidence = 0.8, maxlen = 10;

I want to focus on more rules with lower confidence which is 60% and a low support value 0.01. So I will generate enough rules that I can use them to do my research.

Finally, I got 46 rules.

### 3. What is the lift and conviction of your best and worst rules?

#best rule  
interestMeasure(rules.50.pruned[1],measure= "lift")

## [1] 5.5918

interestMeasure(rules.50.pruned[1],transactions = mydata,measure= "conviction")

## [1] 76.20158

#worst  
interestMeasure(rules.50.pruned[46],measure= "lift")

## [1] 1.045703

interestMeasure(rules.50.pruned[46],transactions = mydata,measure= "conviction")

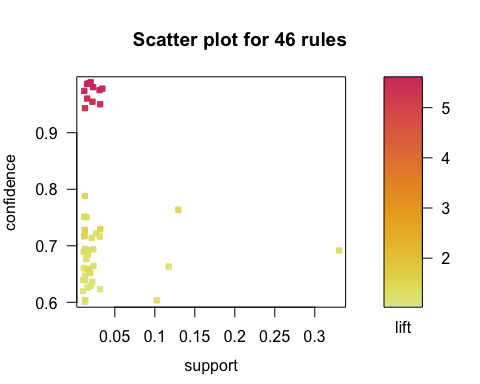
## [1] 1.065849

The lift of best rules is 5.5918; the conviction of best rules is 76.20158;

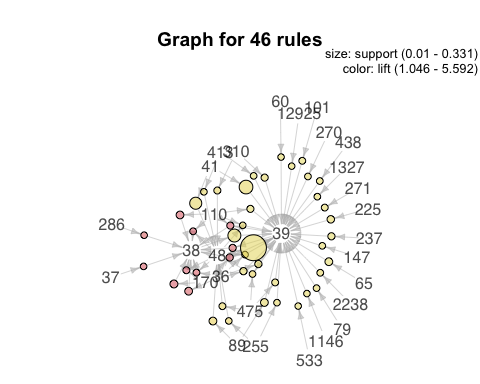
The lift of worst rules is 1.045703; the conviction of worst rules is 1.065849

### 4. Visualize your 50 association rules. Where do the best and worst end up in your plot?

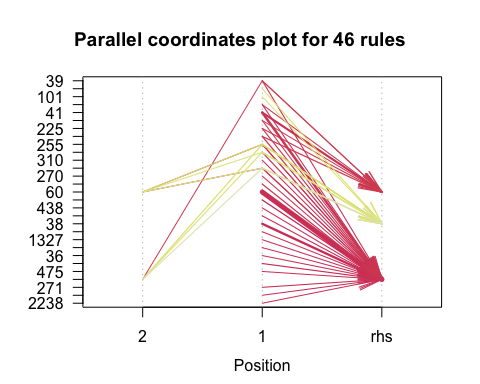
#all rules together  
plot(rules.50.pruned)



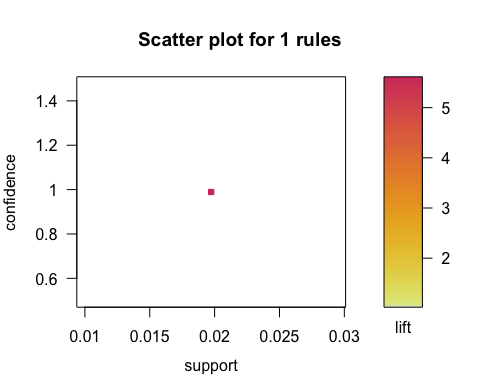
plot(rules.50.pruned, method = "graph", control = list(type="items"))



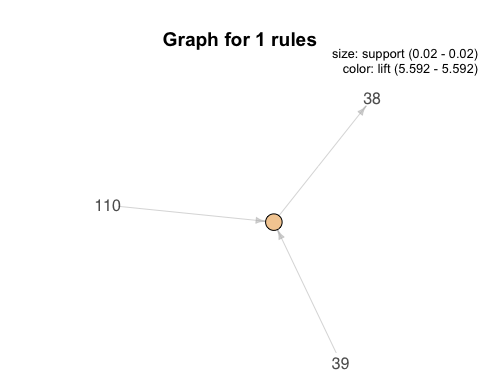
plot(rules.50.pruned, method = "paracoord", control = list(reorder = TRUE))



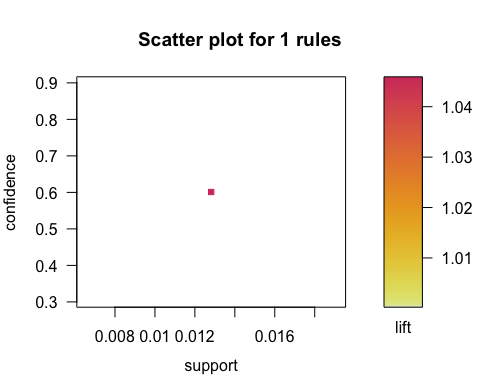
#the best  
plot(rules.50.pruned[1])



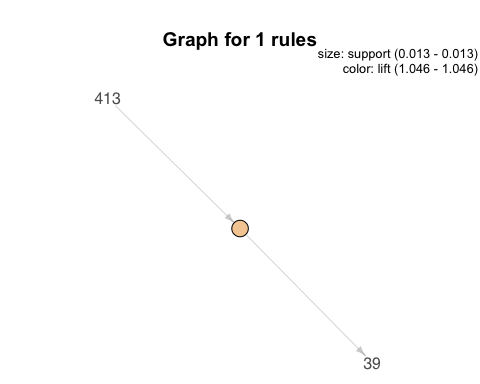
plot(rules.50.pruned[1], method = "graph", control = list(type="items"))



#the worst  
plot(rules.50.pruned[46])



plot(rules.50.pruned[46], method = "graph", control = list(type="items"))



Scatter plot:

The best rule is on the support 0.02, confidence close to 1, lift about 3.6.

The worst ruleis on the support 0.013, confidence close to 0.6, lift 1.025.

Graph:

This graph displays the rule using arrow.

The best rule is shown that item 110 and 39 point to a same cycle and point out to item 38.

The worst rule is shown that item 413 point to the cycle and point out to item 39.

### 5. Does the model make sense?

This model makes sense. This model can figure out the association between different items, and some of the rules have a high confidence as well as lift or support. However, I can't figure out more information of these rules without a better description of the data.