Martin\_Alonso\_Hw3

Martin Alonso

October 20, 2018

# HW3 Instructions

For this homework, we are going to explore the bank data, available on the LMS, and an accompanying description of the attributes and their values. The dataset contains attributes on each person’s demographics and banking information in order to determine they will want to obtain the new PEP (Personal Equity Plan). Your goal is to perform Association Rule discovery on the dataset using Weka.

require(RWeka)

## Loading required package: RWeka

## Error: package or namespace load failed for 'RWeka':  
## .onLoad failed in loadNamespace() for 'rJava', details:  
## call: fun(libname, pkgname)  
## error: JAVA\_HOME cannot be determined from the Registry

require(arules)

## Loading required package: arules

## Loading required package: Matrix

##   
## Attaching package: 'arules'

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

require(arulesViz)

## Loading required package: arulesViz

## Loading required package: grid

require(dplyr)

## Loading required package: dplyr

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:arules':  
##   
## intersect, recode, setdiff, setequal, union

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## Loading the data set, exploring, and cleaning

Before we start doing association rules mining, we need to load the required data set - bankdata\_csv\_all.csv -, explore it, and clean it.

dat <- read.csv("bankdata\_csv\_all.csv", stringsAsFactors = FALSE)  
  
# Check the structure, summary, and head of the data set.  
str(dat)

## 'data.frame': 600 obs. of 12 variables:  
## $ id : chr "ID12101" "ID12102" "ID12103" "ID12104" ...  
## $ age : int 48 40 51 23 57 57 22 58 37 54 ...  
## $ sex : chr "FEMALE" "MALE" "FEMALE" "FEMALE" ...  
## $ region : chr "INNER\_CITY" "TOWN" "INNER\_CITY" "TOWN" ...  
## $ income : num 17546 30085 16575 20375 50576 ...  
## $ married : chr "NO" "YES" "YES" "YES" ...  
## $ children : int 1 3 0 3 0 2 0 0 2 2 ...  
## $ car : chr "NO" "YES" "YES" "NO" ...  
## $ save\_act : chr "NO" "NO" "YES" "NO" ...  
## $ current\_act: chr "NO" "YES" "YES" "YES" ...  
## $ mortgage : chr "NO" "YES" "NO" "NO" ...  
## $ pep : chr "YES" "NO" "NO" "NO" ...

summary(dat)

## id age sex region   
## Length:600 Min. :18.00 Length:600 Length:600   
## Class :character 1st Qu.:30.00 Class :character Class :character   
## Mode :character Median :42.00 Mode :character Mode :character   
## Mean :42.40   
## 3rd Qu.:55.25   
## Max. :67.00   
## income married children car   
## Min. : 5014 Length:600 Min. :0.000 Length:600   
## 1st Qu.:17265 Class :character 1st Qu.:0.000 Class :character   
## Median :24925 Mode :character Median :1.000 Mode :character   
## Mean :27524 Mean :1.012   
## 3rd Qu.:36173 3rd Qu.:2.000   
## Max. :63130 Max. :3.000   
## save\_act current\_act mortgage   
## Length:600 Length:600 Length:600   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##   
## pep   
## Length:600   
## Class :character   
## Mode :character   
##   
##   
##

head(dat)

## id age sex region income married children car save\_act  
## 1 ID12101 48 FEMALE INNER\_CITY 17546.0 NO 1 NO NO  
## 2 ID12102 40 MALE TOWN 30085.1 YES 3 YES NO  
## 3 ID12103 51 FEMALE INNER\_CITY 16575.4 YES 0 YES YES  
## 4 ID12104 23 FEMALE TOWN 20375.4 YES 3 NO NO  
## 5 ID12105 57 FEMALE RURAL 50576.3 YES 0 NO YES  
## 6 ID12106 57 FEMALE TOWN 37869.6 YES 2 NO YES  
## current\_act mortgage pep  
## 1 NO NO YES  
## 2 YES YES NO  
## 3 YES NO NO  
## 4 YES NO NO  
## 5 NO NO NO  
## 6 YES NO YES

Now that we have checked the data set, we’ll start cleaning it. We need to eliminate the id variable and convert the numeric fields into ordinal variables.

# Drop the id column  
datClean <- dat[, 2:12]  
  
# Convert numeric variable to ordinal.  
for (i in 1:ncol(datClean)) {  
 if (class(datClean[[i]]) != "factor") {  
 datClean[[i]] <- as.factor(datClean[[i]])  
 }  
}

With the data converted, we can start exploring different rules.

## Association Rules Mining

# We set the support and confidence parameters to 0.35 and 0.60,  
# respectively, to obtain 42 sets of rules.  
rules <- apriori(datClean, parameter = list(supp = 0.25, conf = 0.7))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.7 0.1 1 none FALSE TRUE 5 0.25 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: 150   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[671 item(s), 600 transaction(s)] done [0.00s].  
## sorting and recoding items ... [16 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 done [0.00s].  
## writing ... [42 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

# Let's explore the rules.  
summary(rules)

## set of 42 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 1 2 3   
## 1 17 24   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 2.000 3.000 2.548 3.000 3.000   
##   
## summary of quality measures:  
## support confidence lift count   
## Min. :0.2517 Min. :0.7011 Min. :0.9645 Min. :151.0   
## 1st Qu.:0.2675 1st Qu.:0.7283 1st Qu.:1.0011 1st Qu.:160.5   
## Median :0.3192 Median :0.7492 Median :1.0175 Median :191.5   
## Mean :0.3438 Mean :0.7490 Mean :1.0276 Mean :206.3   
## 3rd Qu.:0.3729 3rd Qu.:0.7690 3rd Qu.:1.0424 3rd Qu.:223.8   
## Max. :0.7583 Max. :0.8182 Max. :1.2397 Max. :455.0   
##   
## mining info:  
## data ntransactions support confidence  
## datClean 600 0.25 0.7

# We find that the minimum lift is 0.98, while the maximum lift is 1.13,  
# giving us a good set of robust rules, while also throwing in the mix some  
# independent associations. Let's see the top 10 rules that we have found.  
options(digits = 3)  
rules <- sort(rules, decreasing = TRUE, by = "lift")  
inspect(rules[1:10])

## lhs rhs support confidence  
## [1] {mortgage=NO,pep=NO} => {married=YES} 0.285 0.818   
## [2] {save\_act=YES,pep=NO} => {married=YES} 0.292 0.745   
## [3] {pep=NO} => {married=YES} 0.403 0.742   
## [4] {current\_act=YES,pep=NO} => {married=YES} 0.295 0.725   
## [5] {married=YES,pep=NO} => {mortgage=NO} 0.285 0.707   
## [6] {current\_act=YES,pep=NO} => {save\_act=YES} 0.298 0.734   
## [7] {car=NO,mortgage=NO} => {current\_act=YES} 0.263 0.802   
## [8] {car=YES,current\_act=YES} => {save\_act=YES} 0.267 0.727   
## [9] {married=YES,pep=NO} => {save\_act=YES} 0.292 0.723   
## [10] {married=NO} => {current\_act=YES} 0.270 0.794   
## lift count  
## [1] 1.24 171   
## [2] 1.13 175   
## [3] 1.12 242   
## [4] 1.10 177   
## [5] 1.08 171   
## [6] 1.06 179   
## [7] 1.06 158   
## [8] 1.05 160   
## [9] 1.05 175   
## [10] 1.05 162

# The top rule has lift of 1.24, showing that if you don't have a mortgage  
# nor pep, chances are you're likely married. Interesting as the rule might  
# give the idea that these are causes of people getting married.

## PEP analysis

Now, let’s explore what rules can explain whether a customer takes a PEP or not.

datTransactions <- as(datClean, "transactions")  
  
pepRules <- apriori(datTransactions, parameter = list(maxlen = 4), appearance = list(rhs = c("pep=YES",   
 "pep=NO")))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.8 0.1 1 none FALSE TRUE 5 0.1 1  
## maxlen target ext  
## 4 rules FALSE  
##   
## Algorithmic control:  
## filter tree heap memopt load sort verbose  
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE  
##   
## Absolute minimum support count: 60   
##   
## set item appearances ...[2 item(s)] done [0.00s].  
## set transactions ...[671 item(s), 600 transaction(s)] done [0.00s].  
## sorting and recoding items ... [22 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4

## Warning in apriori(datTransactions, parameter = list(maxlen = 4),  
## appearance = list(rhs = c("pep=YES", : Mining stopped (maxlen reached).  
## Only patterns up to a length of 4 returned!

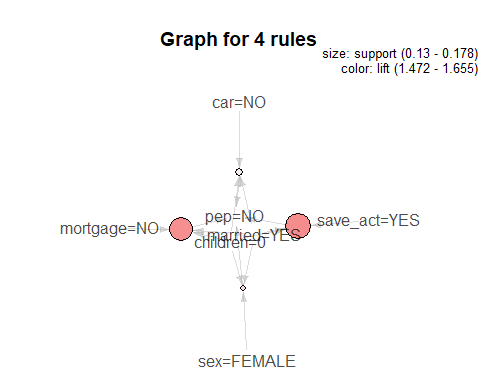
## done [0.00s].  
## writing ... [10 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

inspect(pepRules)

## lhs rhs support confidence lift count  
## [1] {children=1} => {pep=YES} 0.183 0.815 1.78 110  
## [2] {children=1,   
## mortgage=NO} => {pep=YES} 0.118 0.845 1.85 71  
## [3] {married=YES,   
## children=1} => {pep=YES} 0.123 0.831 1.82 74  
## [4] {children=1,   
## save\_act=YES} => {pep=YES} 0.133 0.842 1.84 80  
## [5] {children=1,   
## current\_act=YES} => {pep=YES} 0.140 0.832 1.82 84  
## [6] {children=1,   
## save\_act=YES,   
## current\_act=YES} => {pep=YES} 0.105 0.863 1.89 63  
## [7] {sex=FEMALE,   
## married=YES,   
## children=0} => {pep=NO} 0.130 0.830 1.53 78  
## [8] {married=YES,   
## children=0,   
## car=NO} => {pep=NO} 0.133 0.800 1.47 80  
## [9] {married=YES,   
## children=0,   
## mortgage=NO} => {pep=NO} 0.173 0.897 1.65 104  
## [10] {married=YES,   
## children=0,   
## save\_act=YES} => {pep=NO} 0.178 0.899 1.65 107

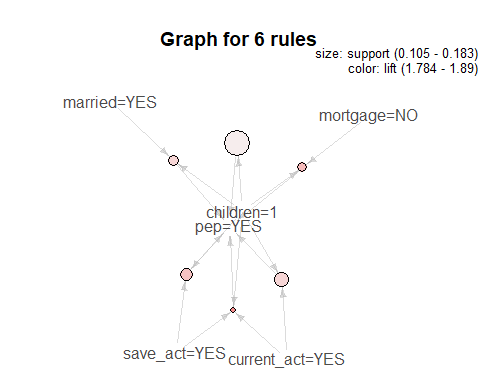
This is very interesting. If we focus solely on the bottom four rules, we find that not only these have pep=NO on the RHS but, on the LHS, they all share the same rule: No children. This can mean that married couples with no children have less expenses, therefore they don’t need a personal equity plan. Similarly, these couples don’t have mortgages, cars, or savings accounts.

plot(pepRules[7:10], method = "graph")



These four rules are very interesting and, given that they all have high lift and confidence levels above 0.80, I would personally advise not to offer Personal Equity Plans to couples who don’t have children.

plot(pepRules[1:6], method = "graph")



However, couple who do have children are likelier to take out a PEP, making them a prime target to offer this product. But, I would also add that, among parents, those who currently have an account plus a savings account, are far more likely to have a PEP than those who don’t. This is clear because, among the set of rules for people who do have a PEP, those who have childern, savings accounts, and other accounts have a confidence of 0.86 and the highest lift out of all the rules with 1.89; meaning that having a PEP is dependent on having the prior three things.

inspect(sort(pepRules[6:10], decreasing = TRUE, by = "lift"))

## lhs rhs support confidence lift count  
## [1] {children=1,   
## save\_act=YES,   
## current\_act=YES} => {pep=YES} 0.105 0.863 1.89 63  
## [2] {married=YES,   
## children=0,   
## save\_act=YES} => {pep=NO} 0.178 0.899 1.65 107  
## [3] {married=YES,   
## children=0,   
## mortgage=NO} => {pep=NO} 0.173 0.897 1.65 104  
## [4] {sex=FEMALE,   
## married=YES,   
## children=0} => {pep=NO} 0.130 0.830 1.53 78  
## [5] {married=YES,   
## children=0,   
## car=NO} => {pep=NO} 0.133 0.800 1.47 80