Miller Chapter 9, Revised with Comments

Rob Carver, March 2017

## Introduction

This script demonstrates two powerful R packages for discovering "association rules" using hypothetical market basket data. There are three packages used here:

* arules -- for discovering the rules
* arulesViz -- for visualization of rules
* RColorBrewer -- an extended color pallete for plotting

The data frame is actually included in package arules and is called Groceries. Market basket data must be in a specific layout for arules to work

This version of the script has been slightly modified. Notes below explain the modifications.

## load the packages

# Association Rules for Market Basket Analysis (R)  
  
library(arules) # association rules  
library(arulesViz) # data visualization of association rules  
library(RColorBrewer) # color palettes for plots  
data(Groceries) # grocery transactions object from arules package

The script then describes the *dimension* of the data matrix.

# show the dimensions of the transactions object

print(dim(Groceries))

## [1] 9835 169

print(dim(Groceries)[1]) # 9835 market baskets for shopping trips

## [1] 9835

print(dim(Groceries)[2]) # 169 initial store items

## [1] 169

head(Groceries)

## transactions in sparse format with  
## 6 transactions (rows) and  
## 169 items (columns)

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 level2 level1  
## 1 frankfurter sausage meat and sausage  
## 2 sausage sausage meat and sausage  
## 3 liver loaf sausage meat and sausage

## Examine frequency for each item with support greater than 0.025

In association analysis, *Support* refers to the proportion of market baskets (orders) containing a particular item.

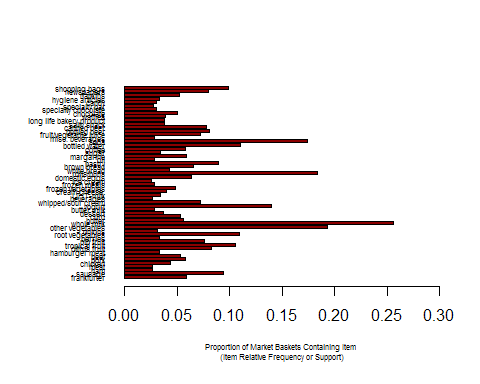
In the Chapter 9 script, Miller places this and other graphs into 8.5 x 11 pdfs. For the purposes of this document, they appear right within the flow of the document.

The function itemFrequencyPlot is part of package arules. The function inspects the Groceries object. For each item in the dataset, it computes the percent of all transactions containing the item, and then selectively displays all items with support > 0.25.

The option cex.names = 0.5 reduces the font sizes for the item names, and xlim = c(0,0.3) limits the x-axis for support values to begin at 0 and stop at 0.3.

las = 1 controls the orientation of the axis label (an admittedly obscure option in r graphics!)

# first set some graphic parameters for upcoming graphs  
plot.new()  
par(pin=c(4, 6)) # set size of plot area   
itemFrequencyPlot(Groceries, support = 0.025, cex.names=0.5, cex.lab=0.5, xlim = c(0,0.3),  
 type = "relative", horiz = TRUE, col = "dark red", las = 1,   
 xlab = paste("Proportion of Market Baskets Containing Item",  
 "\n(Item Relative Frequency or Support)"))



# note that \n splits the axis label into 2 lines

## Explore possibilities for combining similar items

arules defines levels of associations within the data. In this section, Miller examines how many items are in each level.

print(head(itemInfo(Groceries)))

## labels level2 level1  
## 1 frankfurter sausage meat and sausage  
## 2 sausage sausage meat and sausage  
## 3 liver loaf sausage meat and sausage  
## 4 ham sausage meat and sausage  
## 5 meat sausage meat and sausage  
## 6 finished products sausage meat and sausage

print(levels(itemInfo(Groceries)[["level1"]])) # 10 levels... too few

## [1] "canned food" "detergent" "drinks"   
## [4] "fresh products" "fruit and vegetables" "meat and sausage"   
## [7] "non-food" "perfumery" "processed food"   
## [10] "snacks and candies"

print(levels(itemInfo(Groceries)[["level2"]])) # 55 distinct levels

## [1] "baby food" "bags"   
## [3] "bakery improver" "bathroom cleaner"   
## [5] "beef" "beer"   
## [7] "bread and backed goods" "candy"   
## [9] "canned fish" "canned fruit/vegetables"   
## [11] "cheese" "chewing gum"   
## [13] "chocolate" "cleaner"   
## [15] "coffee" "condiments"   
## [17] "cosmetics" "dairy produce"   
## [19] "delicatessen" "dental care"   
## [21] "detergent/softener" "eggs"   
## [23] "fish" "frozen foods"   
## [25] "fruit" "games/books/hobby"   
## [27] "garden" "hair care"   
## [29] "hard drinks" "health food"   
## [31] "jam/sweet spreads" "long-life bakery products"   
## [33] "meat spreads" "non-alc. drinks"   
## [35] "non-food house keeping products" "non-food kitchen"   
## [37] "packaged fruit/vegetables" "perfumery"   
## [39] "personal hygiene" "pet food/care"   
## [41] "pork" "poultry"   
## [43] "pudding powder" "sausage"   
## [45] "seasonal products" "shelf-stable dairy"   
## [47] "snacks" "soap"   
## [49] "soups/sauces" "staple foods"   
## [51] "sweetener" "tea/cocoa drinks"   
## [53] "vegetables" "vinegar/oils"   
## [55] "wine"

# Aggregate items for clearer meaning

Based on the exploration, we aggregate items using the 55 level2 levels for food categories to create a more meaningful set of items.

groceries <- aggregate(Groceries, itemInfo(Groceries)[["level2"]])   
  
print(dim(groceries)[1]) # 9835 market baskets for shopping trips

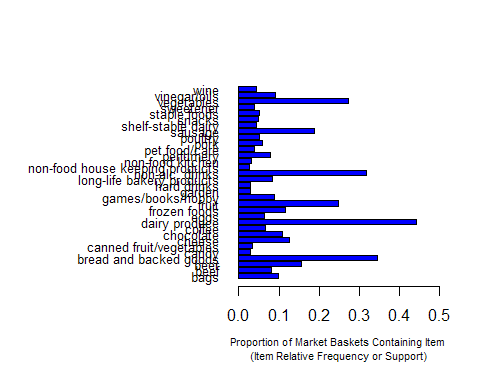
## [1] 9835

print(dim(groceries)[2]) # 55 final store items (categories)

## [1] 55

Next, we look at support for the aggregated item categories

itemFrequencyPlot(groceries, support = 0.025, cex.names=.8,   
 xlim = c(0,0.5), cex.lab=0.7,  
 type = "relative", horiz = TRUE, col = "blue", las = 1,  
 xlab = paste("Proportion of Market Baskets Containing Item",  
 "\n(Item Relative Frequency or Support)"))

 # Rule Sets

obtain large set of association rules for items by category and all shoppers. This is done by setting very low criteria for support and confidence. We need to discuss support and confidence in class.

Rule settting is an iterative process aimed at coming up with a modest list of useful rules. As noted in the comments, the first pass yields nearly 70,000 rules. The second pass -- after adjusting support and confidence thresholds, yields a shorter list.

In reality, we'd keep adjusting the limits until we are satisfied.

first.rules <- apriori(groceries,   
 parameter = list(support = 0.001, confidence = 0.05))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.05 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: 9   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[55 item(s), 9835 transaction(s)] done [0.00s].  
## sorting and recoding items ... [54 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 5 6 7 8 done [0.02s].  
## writing ... [69921 rule(s)] done [0.01s].  
## creating S4 object ... done [0.04s].

print(summary(first.rules)) # yields 69,921 rules... too many

## set of 69921 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 1 2 3 4 5 6 7 8   
## 21 1205 10467 23895 22560 9888 1813 72   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 4.000 4.000 4.502 5.000 8.000   
##   
## summary of quality measures:  
## support confidence lift   
## Min. :0.001017 Min. :0.0500 Min. : 0.4475   
## 1st Qu.:0.001118 1st Qu.:0.2110 1st Qu.: 1.8315   
## Median :0.001525 Median :0.4231 Median : 2.2573   
## Mean :0.002488 Mean :0.4364 Mean : 2.5382   
## 3rd Qu.:0.002339 3rd Qu.:0.6269 3rd Qu.: 2.9662   
## Max. :0.443010 Max. :1.0000 Max. :16.1760   
##   
## mining info:  
## data ntransactions support confidence  
## groceries 9835 0.001 0.05

# select association rules using thresholds for support and confidence   
second.rules <- apriori(groceries,   
 parameter = list(support = 0.025, confidence = 0.05))

## Apriori  
##   
## Parameter specification:  
## confidence minval smax arem aval originalSupport maxtime support minlen  
## 0.05 0.1 1 none FALSE TRUE 5 0.025 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: 245   
##   
## set item appearances ...[0 item(s)] done [0.00s].  
## set transactions ...[55 item(s), 9835 transaction(s)] done [0.00s].  
## sorting and recoding items ... [32 item(s)] done [0.00s].  
## creating transaction tree ... done [0.00s].  
## checking subsets of size 1 2 3 4 done [0.00s].  
## writing ... [344 rule(s)] done [0.00s].  
## creating S4 object ... done [0.00s].

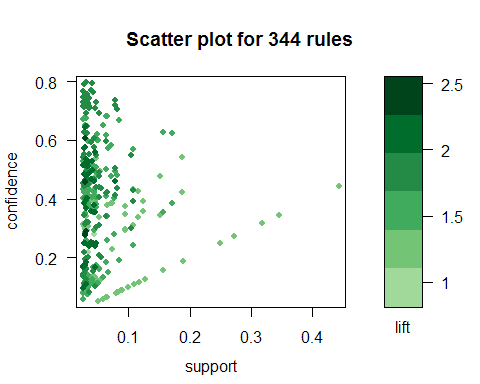
print(summary(second.rules)) # yields 344 rules

## set of 344 rules  
##   
## rule length distribution (lhs + rhs):sizes  
## 1 2 3 4   
## 21 162 129 32   
##   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.0 2.0 2.0 2.5 3.0 4.0   
##   
## summary of quality measures:  
## support confidence lift   
## Min. :0.02542 Min. :0.05043 Min. :0.6669   
## 1st Qu.:0.03030 1st Qu.:0.18202 1st Qu.:1.2498   
## Median :0.03854 Median :0.39522 Median :1.4770   
## Mean :0.05276 Mean :0.37658 Mean :1.4831   
## 3rd Qu.:0.05236 3rd Qu.:0.51271 3rd Qu.:1.7094   
## Max. :0.44301 Max. :0.79841 Max. :2.4073   
##   
## mining info:  
## data ntransactions support confidence  
## groceries 9835 0.025 0.05

# Data visualization of association rules in scatter plot

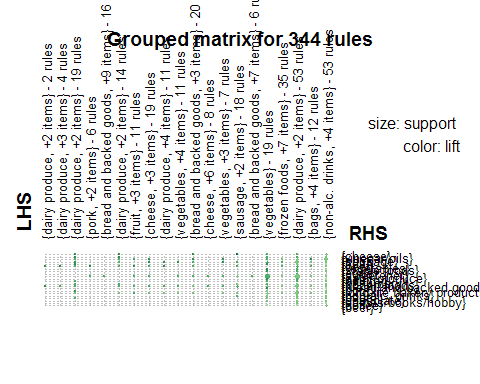
We can represent the rules in a scatter plot and in a matrix.

plot(second.rules,   
 control=list(jitter=2, col = rev(brewer.pal(9, "Greens")[4:9])),  
 shading = "lift")



# grouped matrix of rules

plot(second.rules, method="grouped", cex.names = .7,   
 control=list(col = rev(brewer.pal(9, "Greens")[4:9])))



# select rules with vegetables in consequent (right-hand-side) item subsets

The rules tend to be in the conditional form of "if you buy A, then you are likely to buy B". Sometimes we want to focus on the "consequent" side -- what do people by "before" (say) veggies?

In Miller's code, he prints all 41 rules, then sorts them by life and prints the top 10. We will skip the first printing of all rules.

Note in the code the option `subset = rhs %pin% "vegetables".

vegie.rules <- subset(second.rules, subset = rhs %pin% "vegetables")  
  
  
# sort by lift and identify the top 10 rules  
  
top.vegie.rules <- head(sort(vegie.rules, decreasing = TRUE, by = "lift"), 10)  
inspect(top.vegie.rules)

## lhs rhs support confidence lift  
## [1] {beef,   
## dairy produce} => {vegetables} 0.02989324 0.6074380 2.225010  
## [2] {poultry} => {vegetables} 0.02897814 0.5745968 2.104715  
## [3] {dairy produce,   
## fruit,   
## sausage} => {vegetables} 0.02714794 0.5741935 2.103238  
## [4] {beef} => {vegetables} 0.04585663 0.5595533 2.049612  
## [5] {dairy produce,   
## vinegar/oils} => {vegetables} 0.03141840 0.5355286 1.961610  
## [6] {fruit,   
## sausage} => {vegetables} 0.03426538 0.5290424 1.937852  
## [7] {bread and backed goods,   
## dairy produce,   
## fruit} => {vegetables} 0.04077275 0.5276316 1.932684  
## [8] {pork} => {vegetables} 0.03009659 0.5220459 1.912224  
## [9] {cheese,   
## fruit} => {vegetables} 0.02674123 0.5197628 1.903861  
## [10] {dairy produce,   
## fruit,   
## non-alc. drinks} => {vegetables} 0.03304525 0.5183413 1.898654

# Plot the top 10 veggie rules

plot(top.vegie.rules, method="graph",  
 control=list(type="items"),   
 shading = "lift")

