Lab Exercise 5: Association Rules

Purpose:	This lab is designed to investigate and practice Association Rules. After completing the tasks in this lab you should able to: • Use R functions for Association Rule based models

Tasks:	Tasks you will complete in this lab include: • Use the R –Studio environment to code Association Rule models • Apply constraints in the Market Basket Analysis methods such as minimum thresholds on support and confidence measures that can be used to select interesting rules from the set of all possible rules • Use R graphics "arules" to execute and inspect the models and the effect of the various thresholds

References:	• The groceries data set - provided for arules by Michael Hahsler, Kurt Hornik and Thomas Reutterer. http://rss.acs.unt.edu/Rdoc/library/arules/html/Groceries.html ○ Michael Hahsler, Kurt Hornik, and Thomas Reutterer (2006) Implications of probabilistic data modeling for mining association rules. In M. Spiliopoulou, R. Kruse, C. Borgelt, A. Nuernberger, and W. Gaul, editors, *From Data and Information Analysis to Knowledge Engineering, Studies in Classification, Data Analysis, and Knowledge Organization*, pages 598–605. Springer-Verlag.

Workflow Overview

1. Set the Working Directory and install the “arules” and "arulesViz" package
2. Read in the Data for Modeling
3. Review Transaction data
4. Plot Transactions
5. Mine the Association Rules
6. Read in Groceries dataset
7. Mine the Rules for the Groceries Data
8. Extract Rules with Confidence >0.8

LAB Instructions

Step	Action
1	Log in with gpadmin credentials on to R-Studio. The R Code for this exercise is available at home/gpadmin/LAB05/mba.R

2	**Set the Working Directory and install the packages:**
	To understand Market Basket Analysis and the R package "arules," use a simple set of transaction lists of "book-purchases".
	
	1. Set the working directory to ~/LAB05/ by executing the command:
	
	`setwd("~/LAB05")`
	
	2. Load the package (select the mirror if prompted) and the required libraries:
	
	`#Install the packages and load libraries`
	`>install.packages('arules')`
	`>install.packages('arulesViz')`
	`>library('arules')`
	`>library ('arulesViz')`

Step	Action
3	**Read in the Data for Modeling:** • **Transaction List** is a special data type function in the "arules" package. Read the data in as a Transaction List using the following statement for the states data, "MBAdata.csv". `> txn <- read.transactions ("MBAdata.csv",rm.duplicates = FALSE,format="single",sep=",",cols=c(1,2))` The arguments for the **read.transaction functions** are detailed below: • **file** — the file name. • **format** — a character string indicating the format of the data set. One of "basket" or "single", can be abbreviated. • **Sep** — a character string specifying how fields are separated in the data file, or NULL (default). For basket format, this can be a regular expression; otherwise, a single character must be given. The default corresponds to white space separators. • **Cols** — For the 'single' format, cols is a numeric vector of length two giving the numbers of the columns (fields) with the transaction and item ids, respectively. For the 'basket' format, cols can be a numeric scalar giving the number of the column (field) with the transaction ids. If cols = NULL • **rm.duplicates** — a logical value specifying if duplicate items should be removed from the transactions.

4	**Review Transaction Data:**
	1. First inspect the transaction data
	`>txn@transactionInfo`
	`>txn@itemInfo`
	
	2. Review the results on the console

5	**Plot Transactions:** 1. Use the "image" function that shows a visual representation of the transaction set in which the rows are individual transactions (identified by transaction ids) and the dark squares are items contained in each transaction. `> image(txn)` 2. Review the output in the graphics window

Step	Action

6	**Mine the Association Rules:**
	
	The help(apriori) command provides documentation of the *apriori* function from the *arulesr* package. The syntax of the *apriori* function is as follows:
	
	`rules <- apriori(File, parameter = list(supp = 0.5, conf = 0.9, target = "rules"))`
	
	where the arguments are:
	
	• **data** object of class transactions or any data structure which can be coerced into transactions (for example, a binary matrix or data.frame).
	
	• **parameter** named list. The default behavior is to mine rules with support 0.1, confidence 0.8, and maxlen 5.
	
	1. Build the rules from the transaction data:
	
	`> basket_rules <- apriori(txn,parameter=list(sup=0.5,conf=0.9,target="rules"))`
	
	2. Review the output on the console. The number of rules generated can be seen in the output.
	
	3. Inspect the rule using the following statement:
	
	`> inspect(basket_rules)`
	
	4. Review the output. Observe the generated rule and the support, confidence and the lift thresholds for the rule.

Step	Action

7	**Read in Groceries dataset:**
	
	Use the standard data set, "Groceries" available with the "arules" package.
	
	The Groceries data set contains 1 month (30 days) of real-world point-of-sale transaction data from a typical local grocery outlet. The data set contains 9835 transactions and the items are aggregated to 169 categories.
	
	Read in the data set and inspect the item information
	
	`> data(Groceries)`
	`> Groceries`
	`> Groceries@itemInfo`

8	**Mine the Rules for the Groceries Dataset:** `> rules <- apriori(Groceries, parameter=list(support=0.001, confidence=0.5))` Note the values used for the parameter list. How many rules are generated?

9	**Extract Rules with Confidence >0.8:**
	
	1. Execute the following commands:
	`> subrules <- rules[quality(rules)$confidence > 0.8]`
	`> plot(subrules, control = list(jitter=2))`
	`> inspect(subrules)`
	
	2. Review the results.
	
	3. How many sub-rules did you extract?
	
	4. Extract the top three rules with high threshold for the parameter "lift" and plot.
	
	`> #Extract the top three rules with high lift`
	`> rules_high_lift <- head(sort(rules, by="lift"), 3)`
	`> inspect(rules_high_lift)`
	`> plot(rules_high_lift,method="graph",`
	`control=list(type="items"))`
	
	5. List the rules and the value of the parameters associated with these rules:

End of Lab Exercise