Lab 6: Clustering & Association

Michael Surdek

Southern New Hampshire University

## **LAB Instructions**

|  |  |
| --- | --- |
| **Step** | **Action** |
| 1 | Download the lab files from the Learning Environment:   * Start Here > Assignment Guidelines and Rubrics > Data Files * MBAdata (CSV file) * Mba.R (R File) |
| 2 | **Set the Working Directory and install the “arules” package:**  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 <YOUR DIRECTORY> by executing the command:   **setwd("<YOUR DIRECTORY>")**   * (Or using the “Tools” option in the tool bar in the RStudio environment.)  1. Load the package (select the mirror if prompted) and the required libraries:   #**Install the packages and load libraries**  **>install.packages('arules', dependencies = TRUE)**  **>install.packages('arulesViz, dependencies = TRUE ')**  **>library('arules')**  **>library ('arulesViz'**) |
| 3 | **Read in the Data for Modeling:**   * **Transaction List** is a special data type function in the “arules” package.  1. Read the data in as a Transaction List using the following statement for the states data, “MBAdata.csv”.   **> #read in the csv file as a transaction data**  **> 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 (this can vary per version of R)   **>txn@transactionInfo**  **>txn@itemInfo**  Or  **>txn@itemsetInfo**  **>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 |
| 6 | **Mine the Association Rules:**  The **“apriori” function**, provided by the *arulesr* package, is used 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. Read in the statement for the transaction data:  **> #mine association rules**  **> 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 and is represented as follows:  **writing ... [1 rule(s)] done [0.00s]**  3. Inspect the rule using the following statement:  **> inspect(basket\_rules)**  4. Review the output.  5. State the generated rule and the support, confidence and the lift thresholds for the rule  {R-basics} => {Stat-intro}  Support: 0.5714286  Confidence: 1  Lift: 1.166667 |
| 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.   1. Read in the data set and inspect the item information  **> #Read in Groceries data**  **> data(Groceries)**  **> Groceries@itemInfo** |
| 8 | **Mine the Rules for the Groceries Data:**  **> #mine rules**  **> rules <- apriori(Groceries, parameter=list(support=0.001, confidence=0.5))**   * Note the values used for the parameter list.  1. How many rules are generated?   5668 rules |
| 9 | **Extract the Rules in which the Confidence Value is >0.8 and high lift:**  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?   * These rules are more valuable for the business.   371 sub-rules  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:  {Instant food products,soda} => {hamburger meat}  Support: 0.001220132  Confidence: 0.6315789  Lift: 18.99565  {soda,popcorn} => {salty snack}  Support: 0.001220132  Confidence: 0.6315789  Lift: 16.69779  {flour,baking powder} => {sugar}  Support: 0.001016777  Confidence: 0.5555556  Lift: 16.40807 |

*End of Lab Exercise*