Association Rules

Lab 3: Association Rules

- Generation of Frequent Itemsets
- Rule Generation
- Vizualization

Please send comment to (version 1.0 – January 11, 2017):

Romain Billot *romain.billot@telecom-bretagne.eu*
Yannis Haralambous *yannis.haralambous@telecom-bretagne.eu*
Philippe Lenca *philippe.lenca@telecom-bretagne.eu*
Sorin Moga *sorin.moga@telecom-bretagne.eu*

1 Data presentation

Association rules aim at analyzing large collections of transactions, in which each transaction is composed of one or more items. The goal is then to see what items are frequently bought together and to discover a list of rules that describe the purchasing behavior and hence relationships between the items. These relationships depend both on the business context and the nature of the algorithm being used for the discovery. Association rules can be implemented for the purpose of market basket analysis, enabling, for instance, cross-merchandising between products and high-margin or high-ticket items. Moreover, recommender systems can also use association rules to discover related products or identify customers who have similar interests. For example, association rules may suggest that those customers who have bought product A have also bought product B, or those customers who have bought products A, B, C, are more similar to this customer.

–The Grocery dataset– The Groceries dataset has been collected from 30 days of real-world transactions of a grocery store. It contains 9835 transactions and the items are aggregated into 169 categories. The installation of the R *arules* package is required. The class of the dataset is *transactions*, as defined by the *arules* package. The *transactions* class contains three slots:

Slot (not variable)	Meaning
`transactionInfo`	Data frame with vectors of the same length as the number of transactions
`itemInfo`	Data frame to store item labels
`data`	Binary incidence matrix that indicates which item labels appear in every transaction

Table 1: Structure of the Groceries dataset from class *transactions*

Question 1

Load, understand and describe carefully the dataset into R. Use the *data*, *class*, *summary* and the @ character to access the slots listed previously.

2 Generation of Frequent Itemset

In this section, we are going to illustrate, step by step, the implementation of the *Apriori* algorithm to create frequent itemsets.

Question 2

We first assume that the minimum threshold is set to 0.02 based on management discretion. Hence an itemset should appear at least 198 times to be considered as a frequent itemset. The first iteration of the algorithm computes the support of each product in the dataset and retains those products that satisfy the minimum support. Use the *apriori* algorithm and set the $minlen = 1$ and $maxlen = 1$ in the parameter list in order to show 1 itemsets only.

Question 3

Use the *inspect* function to display the top 10 frequent 1-itemsets sorted by their support.

Question 4

In this question, the goal is to compute the support of each candidate 2-itemset and retain those that satisfy the minimum support. Use the *apriori* function again. Do the same step by step with 3-itemsets and 4-itemsets. Conclude.

Question 5

Now run the algorithm without specifying the `maxlen` parameter. The algorithm continues each iteration until it runs out of support or until k reaches the default value of `maxlen` set to 10. What can you conclude ? Try to run the algorithm with other minimum support thresholds.

3 Rule generation and visualization

Question 6

The *apriori* function can also be used to generate rules. Use it by assuming here that the minimum support threshold is 0.001, the minimum confidence threshold 0.6. How many rules are created? What is the impact of both thresholds on the number of generated rules?

Question 7

Display a scatterplot of the 2918 rules. Access to the quality of the rules (with @) to display also a scatterplot matrix that will compare the support, confidence and lift of the 2918 rules

Question 8

Re-use a previous function to show the top ten rules sortest by the lift. Which rule has the highest lift ? Then, create an object that contains the rules whose confidence is above 0.9. How many rules does this object contain?

Question 9

We denote as "antecedent" the left-hand side of a rule and as "consequent" the right-hand side of a rule. Tune the parameters of the plot function in order to create a matrix-based visualization of the consequents versus the antecedents. As a color legend, put a color matrix indicating the lift and the confidence to which each square in the main matrix corresponds.

Question 10

Prepare a graph visualization of the top five rules with the highest lift.