
1 Hands On: Association Rules

1.1 Association Rules

1. Load the packages `arules`, `arulesViz` and the dataset `Groceries` from the package `arules` which contains 1 month of real-world point-of-sale transaction data from a typical local grocery.

- (a) Type `Groceries` on the R prompt. What does it return? Use the function `class` to inspect the type of data set.
- (b) Use the function `summary` to get more information on the data set.
- (c) Use the function `size` on the data set. What information does it return?
- (d) Use the function `inspect` to see the first five transactions.
- (e) Are there any duplicated transactions? Use the function `unique` or `duplicated`.
- (f) Use the function `itemFrequency` to see the relative frequency of each item.
- (g) Using the function `itemFrequencyPlot`, plot the top 5 more frequent items.
- (h) Using the same function `itemFrequencyPlot`, plot the items that have a support value of at least 0.1. How many are there?
- (i) Using function `apriori`, and without generating any rules, obtain the frequent itemsets for a minimum support of 0.01. What is the class of the object returned? How many frequent itemsets were found?
- (j) Inspect the 5 most frequent itemsets. What's their size?
- (k) From the frequent itemsets obtained, select the subset of closed frequent itemsets and the subset of maximal frequent itemsets. What can you conclude?
- (l) Use the function `apriori` to generate association rules from the `Groceries` data set. What is the class of the returned object? How many rules were generated?
- (m) Change the values of minimum support and minimum confidence and see how does that affect the number of rules generated.
- (n) Obtain the association rules with `minsup=0.01` and `minconf=0.25`. Using the functions `summary`, `quality`, `plot` and `inspect` acquire more information on the generated rules.
- (o) Select the rules with a lift value above 2. Use the function `subset` for that.
- (p) Using one instruction only, select the rules that have lift value above 2 and the items "whole milk" or "yogurt" on the consequent. Inspect the selected rules by decreasing order of their lift value.

2. Read the csv file of **German Credit dataset** into a data frame in R. This data set has the record of 1000 persons who took a credit by a bank.
- (a) Remove the first attribute from the data frame, it is just an identifier for each record.
 - (b) Try to convert the data frame into a transactions data set using the function `as`. What do you obtain?
 - (c) Use the function `cut` to discretize the numerical attributes according to the following:
 - `duration_in_month`: 4 equal-width intervals with labels "short", "med-short", "med-long", "long";
 - `credit_amount`: 4 equal-width intervals with labels "small", "med-small", "med-high", "high";
 - `age`: 4 equal-width intervals with labels "young adult", "adult", "senior", "golden".
 - to the rest of numerical attributes, simply use the function `as.factor`
 - (d) Convert the data frame into a data set of transactions. What do you obtain? Use the function `itemInfo` to see what each item represents.
 - (e) Run `apriori` to obtain the association rules from the data set. Plot the obtained rules.
 - (f) Observe the effect of filters and measures on the number of rules generated.
 - (g) Select the rules with confidence equal to 1. What those rules tell you?
 - (h) Run `apriori` again, but this time imposing a minimum confidence equal to 0.6, minimum length of 2 and focusing only on attributes sex, age, job, housing and purpose of credit.
 - (i) Identify rules $a \rightarrow b$ and $b \rightarrow a$. What is that their quality values tell you?
 - (j) Run `apriori` to obtain rules that relate the purpose of credit with age, job and housing. Impose a minimum support of 0.05, minimum confidence of 0.25 and a minimum length of 2. Could you propose a marketing campaign from the obtained rules?
 - (k) Plot the previous set of rules using the method `graph` and `graph` with `itemsets`. What these graphs tell you?
 - (l) Plot the previous set of rules using the method `grouped`.