CS 422: Data Mining

Department of Computer Science
Illinois Institute of Technology
Vijay K. Gurbani, Ph.D.

Fall 2024: Homework 6 (10 points)

Due date: Wednesday November 13 11:59:59 PM Chicago Time

Please read all of the parts of the homework carefully before attempting any question. If you detect any ambiguities in the instructions, please let me know right away instead of waiting until after the homework has been graded.

1. Exercises (3 points divided evenly among the questions) Please submit a PDF file containing answers to these questions. Any other file format that can be read will lead to a loss of 0.5 point. Non-PDF files that cannot be opened and read for grading will lead to a loss of all points allocated to this exercise.

1.1 Tan, Ch. 5 (Association Analysis)

Questions 15

1.2 Zaki, Chapter 8 (Frequent Pattern Mining)

Questions 1(a), 4

2. Practicum problems (7 points)

Association Analysis

In this assignment you will be using the *Extended Bakery* dataset, which describes transactions from a chain of bakery shops that sell a variety of drinks and baked goods.

The dataset is presented as a set of 20,000 transactions, stored in a file named tr-20k.csv. The file contains the data in a sparse vector format, i.e., each line of the file has the following format:

1, 7, 15, 44, 49
2, 1, 19
...

The first column is the transaction ID and the subsequent columns contain a list of purchased goods from the bakery represented by their product ID code. In the example above, the first line implies that transaction ID one contained four items: 7, 15, 44, and 49. The mapping of the product ID to product name is provided in the products.csv file.

(a) **[1 points]** For the tr20k.csv file create a canonical representation of the transaction file. A canonical representation for each dataset will be a file that contains a list of product names (not IDs) on a line, each product separated by a comma and a newline ends the line. So, as an example, the first two lines shown above (7, 15, 44, 49; and 1, 19) would correspond to the following canonical representation, respectively:

Coffee Eclair, Blackberry Tart, Bottled Water, Single Espresso
Lemon Cake, Lemon Tart
…

Save the canonical representation in a new file with the canonical suffix, i.e., tr-20k-canonical.csv. Use this file for the rest of the work. **Include this file in the archive that you upload to Canvas.**

You can use any programming language of your choice to do part (a).

(b) Read the tr-20k-canonical.csv file into memory as a transaction dataset. Based on the transaction dataset you read in, please answer the following questions. To familiarize yourself with the R apriori package APIs, please see https://cran.r-project.org/web/packages/arules/arules.pdf.

(i) **[0.33 points]** How many unique items are there in the dataset? Your function should return a scalar.

(ii) **[0.33 points]** What are the **two most frequent items** in the dataset. Your function should return a named vector, where the name of the vector element is the item and the value is the item's frequency count.

(iii) **[0.33 points]** What are the **two least frequent items** in the dataset? Your function should return a named vector, where the name of the vector element is the item and the value is the item's frequency count.

(iv) **[0.33 points]** How many frequent itemsets are there with a support of 0.01? Your function should return an object of class *itemset*.

(v) **[0.33 points]** How many frequent itemsets are there with a support of 0.10? Your function should return an object of class *itemset*.

(vi) **[1.02 points]** Looking at the number of itemsets retrieved between b-(iv) and b-(v), why do you think there is such a discrepancy in the number of itemsets retrieved? Please explain your answer in no more than 1-2 sentences. Put the answer to this question in the PDF you submit. Label the answer with "Q2(b)(vi) ..."

(vii) **[0.33 points]** If we were to set the support to 0.001, how many itemsets are retrieved? Your function should return an object of class *itemset*.

(c) We now look at generating rules using the apriori() API.

(i) **[0.20 points]** Set the *support* parameter to 0.001 and mine the rules in the dataset. Your function should return an object of class *rules*.

Caution: Do not try to get the number of rules at a support value of 0.0; this will slow your computer down mining the rules to the point it will be unresponsive, and you may have to hard-reboot it.

(ii) **[1.5 points]** Comment on how good do you think the rules are as a function of confidence and lift. If the rules are good, state so and provide a 1-sentence answer why you consider the rules to be good. If you think the rules are not good, state so and provide a 1-sentence answer why you consider the rules to be bad. Put the answer to this question in the PDF you submit. Label the answer with "Q2(c)(ii) ..."

(iii) **[0.65 points]** Recall the top 2 items from b-(ii). What percentage of rules you retrieved in c-(i) contain the most frequent two items identified in b-(ii)? (Hint, use the subset() API in arules package.) Your function should return an object of class *rules*.

(iv) **[0.65 points]** Perform the same analysis as above for the least frequent two items from b-(iii).Your function should return an object of class *rules*.