Homework 2

CS 870

Due: 13th Aug 11:59PM. No late submissions will be allowed. Please start early.

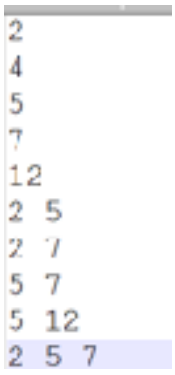
1. This question is on frequent itemset mining. Implement the Apriori Algorithm to mine frequent itemsets. Apply it on the Dataset: http://fimi.ua.ac.be/data/retail.dat. Details about the dataset can be found at http://fimi.ua.ac.be/data/retail.pdf. You may assume all items are integers.

    a. Please name your file RollNo_1a.sh. For example, if MCS162913 is your roll number, your file should be named MCS162913.sh. Executing the command "sh RollNo.sh retail.dat X" should generate a file RollNo.txt containing the frequent itemsets at >=X% support threshold. Notice that X is in percentage and not absolute count. *(30 points)*

    RollNo.txt should strictly follow the following format since evaluation will be through an automated script.

    i. Each frequent itemset must be on a new line. The items must be space separated and in ascending order of ASCII code.

    ii. The itemsets must be grouped based on cardinality and then ordered within in numeric ascending order. For example, if (2), (4),(5),(7),(12),(2,5),(2,7),(5,7), (5,12),(2,5,7) are the frequent itemsets, your output should be.



    b. Compare the performance of your implementation with FP-tree (you are free to use an available implementation here. But the FP-tree implementation and your Apriori implementation must be in the same language) you vary the

        i. Generate a plot where x axis is the support threshold and y axis is the running time. Plot the running times at support thresholds of 1%, 5%, 10%, 25%, 50%, and 90%. Explain the results that you observe. (10 points)

        ii. Generate a plot where x axis is the dataset size in terms of number of transactions at 10%, 25%, 50% and 100% of the entire dataset and y axis is the running time at 10% support threshold. Explain the results that you observe. (10 points)

c. Efficiency Competition: We will have a competition among all submitted implementations of the Apriori algorithm. The fastest would get full points. If your algorithm is X% slower than the fastest, then you would get X% of full points. You would be in this competition only if you get full points in part (a), i.e., you have the correct implementation of the Apriori algorithm. (20 points)

Files you need to submit

- For part a, submit the sh script and all codes that the script calls. Your grade will be (F-score)*30.
- For b, you need to submit a written report explaining your answers. You should also point to the FP tree library you have used. If you choose to implement FP tree, upload the code.
- For c, you don't need to submit anything. Your grade will be fastestTime/ yourTime *20. FastestTime is the fastest submission we receive.

Plagiarism Policy:

Do not copy code from your friend or from the internet. We have downloaded all available libraries of Apriori algorithm from the web and all submitted codes will be checked against these as well as those submitted in this homework. Any plagiarized code will result in an F grade for the course straight-away.