Name: Yutong He

Net ID: yhe29

Student ID: 29433162

E-MAIL: yhe29@u.rochester.edu

Course: CSC 240

DATE: Oct.24th, 2016

Project 1

ALGORITHMS I USE

1. For Apriori, I use the algorithm provided in the textbook
2. For FP-Growth, I use the algorithm provided in the textbook
3. For my improved Apriori, I use the Hash Table and Transaction Scan Reductions strategies
4. I use hash table to store the item ID and the count so that when examining the candidates, it only needs to check the hash table and this will make the process much quicker
5. After generating frequent itemsets for each length-n patterns, I check transactions that do not contain any frequent n-itemsets and remove them because they cannot contain any frequent (n - 1)-itemsets

Data Structures

1. Apriori:
2. String for every item
3. ArrayList<String> for every transaction and patterns
4. ArrayList of ArrayList<String> (i.e. ArrayList<ArrayList<String>>) for database, candidates and pattern set
5. Set for length-1 itemset
6. FP-Growth:
7. String for every item
8. ArrayList<String> for every transaction, patterns and header table
9. ArrayList of ArrayList<String> (i.e. ArrayList<ArrayList<String>>) for database, conditional pattern base and combinations
10. Specified objects for FPTree and header node in the header table
11. Set for frequent patterns
12. HashTable for length-1 itemset
13. Apriori Improved
14. String for every item
15. ArrayList<String> for every transaction and patterns
16. ArrayList of ArrayList<String> (i.e. ArrayList<ArrayList<String>>) for database, candidates and pattern set
17. HashTable for length-1 itemset

RUNNING TIME ANALYSIS

Apriori: 99.634s

FP-Growth: 0.293s

Apriori Improved: 0.538s

Apriori is slow because it has to consistently scan the database and does not have a place to store temporary information.

FP-Growth is faster because it only has to scan the database twice (not including reading in the file) and has FP-Tree and header table to traverse the FP-Tree in the process of generating frequent patterns.

Apriori Improved is faster because it has a header table to refer when generating candidates and also can scan smaller database because of Transaction Scan Reductions.