Data Mining Report

(Assignment - 1)

Group: 8

1. M Ratna Abhishek (S20150020027)
2. G Rambabu (S20150010017)

Goal of the Assignment:

Implement FP Growth algorithm in ANSI C.

Files:

1. FP_Growth.c - Contains the implementation of the FP-Growth algorithm in C.
2. groceries_subset.csv - Input file(data).
3. Images/screenshots:
    a. min_sup_count_20.png(3 files)
    b. min_sup_count_50.png
    c. min_sup_count_60.png
    d. min_sup_count_100.png

Implementation details:

- We wrote different functions for each step. Some of the functions in order are:
    1. Read and store data.
    2. Store unique items and frequencies.
    3. Sort items based on frequency.

4. Remove items from the transactions based on minimum support count.
5. Update transactions using remaining items.
6. Construct tree using the updated transactions.
7. Check multiple paths in the tree.
8. If there are multiple paths in tree, repeat the same process by conditioning on the item.

- There are some other simple functions that are used here and there during the algorithm. Combination function is used while printing frequent item sets.
- We have followed the above 8 steps/functions in the code. The code is well documented (at functions level).
- The function and variable names are easily understandable.

Other details:

- We have defined MACROS for most of the things for easy usability of code.
- Some of the MACROS that are defined in the code are:
    - NUMBER_OF_TRANSACTIONS 1000
    - NUMBER_OF_ITEMS 300
    - MAX_TRANSACTIONS_IN_A_ROW 50
    - CHAR_LENGTH 30 //Max string length of an item.
    - MIN_SUPPORT_COUNT 100
    - FILENAME "groceries_subset.csv"
- These are default parameters used right now in the code.
- Please change the MACROS based on your input file and your need. Otherwise, you may end with “Segmentation fault” error.

OS and Compiler details:

1. **OS:** Ubuntu 16.04
2. **Compiler:** 5.4.0 (gcc)

- The attached code is working fine on the above mentioned version details of Operation System and Compiler.

How to run the code:

1. Go to the folder in which the Assignment code is present.
2. Use the below commands:
    a. gcc FP_Growth.c → A file will be generated.
    b. ./a.out

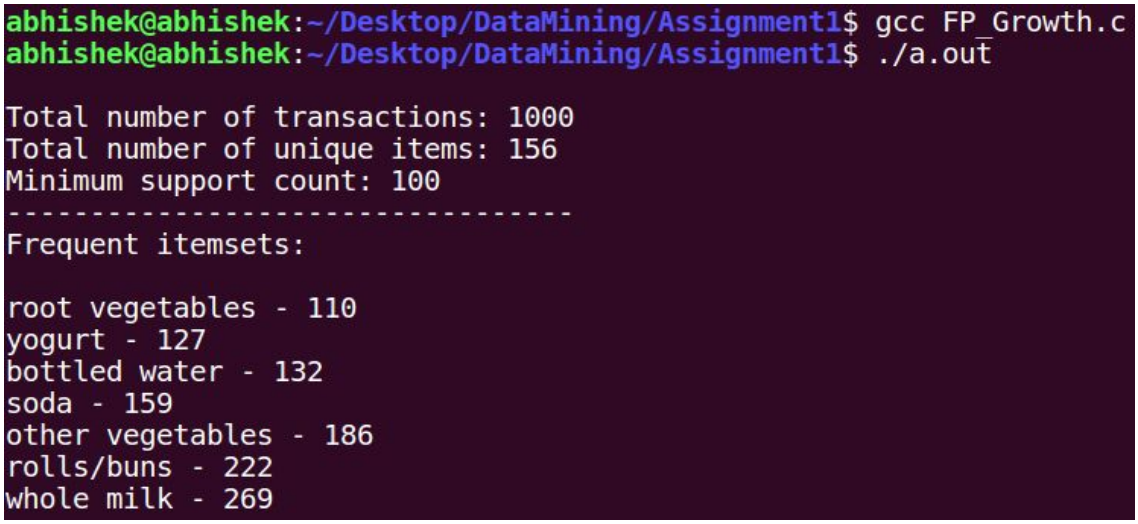
Output format:

1. It displays basic statistics like:
    a. Total number of transactions.
    b. Total number of unique items.
    c. Minimum support count.
2. It prints all the frequent itemsets one by one with their frequencies.

- I have attached some of the screenshots which contains the output that was ran on “groceries_subset.csv” with different minimum support counts.

Output screenshots:

1. Minimum support count - 100



2. Minimum support count - 60

```
abhishek@abhishek:~/Desktop/DataMining/Assignment1$ gcc FP_Growth.c
abhishek@abhishek:~/Desktop/DataMining/Assignment1$ ./a.out

Total number of transactions: 1000
Total number of unique items: 156
Minimum support count: 60
---------------------------------
Frequent itemsets:

fruit/vegetable juice - 67
beef - 69
frankfurter - 73
whipped/sour cream - 74
curd - 75
canned beer - 76
bottled beer - 76
coffee - 76
shopping bags - 78
sausage - 78
pastry - 78
newspapers - 89
citrus fruit - 95
tropical fruit - 97
root vegetables - 110
yogurt - 127
bottled water - 132
soda - 159
other vegetables - 186
other vegetables, whole milk - 72
rolls/buns - 222
rolls/buns, whole milk - 68
whole milk - 269
```

3. Minimum support count - 50

```
abhishek@abhishek:~/Desktop/DataMining/Assignment1$ gcc FP_Growth.c
abhishek@abhishek:~/Desktop/DataMining/Assignment1$ ./a.out

Total number of transactions: 1000
Total number of unique items: 156
Minimum support count: 50
---------------------------------
Frequent itemsets:

brown bread - 50
domestic eggs - 53
pork - 53
margarine - 56
fruit/vegetable juice - 67
beef - 69
frankfurter - 73
whipped/sour cream - 74
curd - 75
canned beer - 76
bottled beer - 76
coffee - 76
shopping bags - 78
sausage - 78
pastry - 78
newspapers - 89
citrus fruit - 95
tropical fruit - 97
root vegetables - 110
yogurt - 127
yogurt, whole milk - 56
bottled water - 132
soda - 159
other vegetables - 186
other vegetables, whole milk - 72
rolls/buns - 222
rolls/buns, whole milk - 68
whole milk - 269
```

- These are some of the outputs that are generated using different minimum support counts.