Data Mining Assignment 1

Group no. 18

Group Members:

Subhadeep Dash(S20160010021)

Ventrapragada Sai Rathan(S20160010104)

Aim: To find out the frequent itemsets in the dataset given using FP trees.

Description: Frequent itemsets can be found out by using the following 3 methods:

1. **Naïve method** (Database scan: 1, Space Complexity: O(2^n), Time Complexity: O(m)), where m is no. of transactions and n is the no. of distinct elements present.

2. **Apriori method** (Database scan: k, Space Complexity & Time Complexity: depends on number of candidates generated), where k is the size of the largest itemset possible.

3. **FP (Frequent Pattern) growth method** (Database scan: 1, Space Complexity and Time complexity depends on the unique elements in the header table).

Among the above-mentioned methods, FP tree is considered to be efficient as Apriori and Naïve algorithms have higher space complexity than FP growth algorithm.

Procedure:

1. Generate the header table with all the elements sorted in decreasing order of their frequencies which have support greater than the specified value.

2. After constructing the header table, find the frequent patterns in the transaction table in order of their decreasing frequencies based on their presence in the header table.

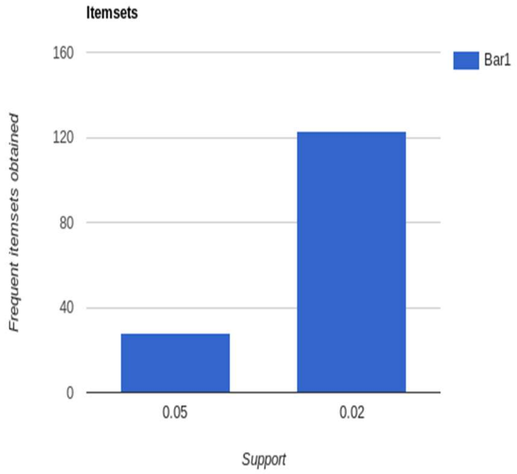
3. Then insert the patterns in a FP tree with the header elements' pointers being pointed to all the elements having same value which helps in getting all the common prefixes more easily for the next recursive step.

4. Keeping the condition of an item being present in the present step, go to the next level of recursion sending the new transaction table, header table and frequent patterns which contain the common prefixes, to the construction function again.

5. These steps keep on repeating until a single element is left out in the FP growth tree.

Statistics:

Comparision of numbers of frequent itemsets obtained with support 0.02 and 0.05 is shown.



Conclusion:

We observe that we gain all the frequent itemsets of any size within a fraction of seconds by using the FP tree data structure which is found to be less hectic than the Apriori algorithm in which we have to generate candidates and also the naïve method which needs a lot of memory.