ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

Comparing the Performance of Frequent Itemsets Mining Algorithms

Kalash Dave[1], Mayur Rathod[2], Parth Sheth[3], Avani Sakhapara[4]

UG Student, Dept. of I.T., K.J.Somaiya College of Engineering, Mumbai, India[1]

UG Student, Dept. of I.T., K.J.Somaiya College of Engineering, Mumbai, India[2]

UG Student, Dept. of I.T., K.J.Somaiya College of Engineering, Mumbai, India[3]

Assistant Professor, Dept. of I.T., K.J.Somaiya College of Engineering, Mumbai, India[4]

ABSTRACT: Frequent Itemset mining is an important concept in Data Mining. With the development of complex applications, huge amount of data is received from the user and collectively stored. In order to make these applications profitable, the stakeholders need to understand important patterns from this data which occur frequently so that the system can be modified or updated as per the evaluated result. The business now-a-days being fast paced, it is important for the frequent itemset mining algorithms to be fast. This paper compares the performance of four such algorithms viz Apriori, ECLAT, FPgrowth and PrePost algorithm on the parameters of total time required and maximum memory usage.

KEYWORDS: Frequent Itemset Mining; Data Mining; Apriori; FPgrowth; PrePost; ECLAT

I. INTRODUCTION

Data mining, or knowledge discovery, is the computer-driven process of searching through and analysing enormous data and then understanding the meaning of the data. Data mining helps predict future trends which allow businesses to make knowledge-driven decisions. Data mining algorithms search databases for hidden patterns, finding useful information that experts may miss because it lies outside their expectations. Companies in a wide range of industries like retail, finance, heath care, manufacturing, transportation, and aerospace are already using data mining tools and techniques to take advantage of historical data.

For businesses, data mining is used to discover relationships in the data to help make better business decisions. Data mining can help spot sales of shares, create better marketing campaigns, and know customer loyalty. In Healthcare industries, Data mining can be used to predict the occurrence of disease by mapping the frequently occurring symptoms amongst its patients.

II. BACKGROUND

A. *Apriori Algorithm:*

The Apriori Algorithm was proposed by Agrawal et.al. in 1994[1]. Apriori is an influential algorithm in market basket analysis for mining frequent items sets for association rules. The algorithm is called Apriori because it requires a prior knowledge of frequent itemset properties. Apriori algorithm requires a number of scanning over the database. The algorithm is as follows in the original paper [1]:

```
$L_1$ = {frequent items};
for ( k = 1; $L_k$ != ∅; k++) do begin
    $C_{k+1}$ = candidates generated from $L_k$;
    for each transaction t in database do increment the count of all candidates in $C_{k+1}$ that are contained in t
    $L_{k+1}$ = candidates in $C_{k+1}$ with min_support
end
```

```
return $U_k$ $L_k$;
```

During pass k, the algorithm finds the set of frequent itemsets L_k of length k that satisfy the minimum support requirement. The algorithm terminates when L_k is empty. A pruning step eliminates any candidate, which has a smaller subset.

B. *ECLAT Algorithm:*

The ECLAT (Equivalence Class Transformation) algorithm uses the depth first search approach to find the elements from the bottom [2]. This algorithm uses vertical database instead of the basic horizontal database. Unlike Apriori algorithm, the ECLAT algorithm scans the database only once [2][4]. The algorithm as mentioned in the paper by M.Zaki [2]:

```
Input: $F_k$ = {$I1$, $I2$,...,$In$} // cluster of frequent k-itemsets.
Output: Frequent $l$-itemsets, $l > k$.
Bottom-Up ($F_k$ ) {
    for all $Ii$ $F_k$
        $F_k$ +1= Φ;
        for all $Ij$Є$F_k$, $i < j$
            $N = Ii \cap Ij$;
            if $N .sup \geq min\_sup$ then
                $F_k$+1 = $F_k$ +1U$N$;
            end
        end
    end
    if $F_k$+1 ≠ Φ then
        Bottom-Up ($F_k$ +1);
end }
```

In this algorithm, F_k stores the number of items as input. Output contains the itemsets which frequently occurred. First F_k +1 is to be considered as the empty database. In the next step find the support of individual items. Now compare the support of Items with the minimum threshold support. Put all those items in F_k +1. F_k +1 contains all frequent items. Again check that F_k +1 is empty or not. If it is not empty then bottom up approach will apply on F_k +1.

C. *FPgrowth Algorithm:*

FP growth algorithmic program is an efficient algorithm for producing the frequent itemsets without generation of candidate itemsets. It adopts a divide and conquer strategy [5] and it needs two database scans to seek out the Support count. The algorithm is mentioned in the paper by author as [5]:

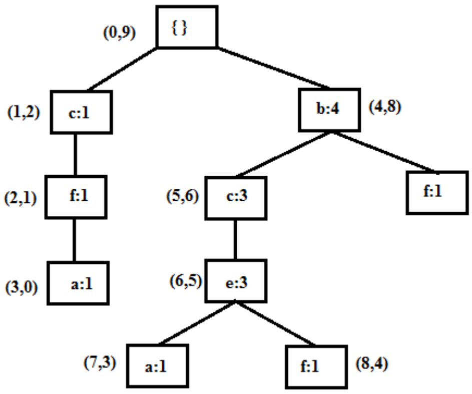
```
Input: A database DB, represented by FP-tree constructed and a minimum support threshold
Output: The complete set of frequent patterns.
Method: call FPgrowth (FP-tree, null).
Procedure FPgrowth (Tree, a) {
    if Tree contains a single prefix path then // Mining single prefix-path FP-tree {
        let P be the single prefix-path part of Tree;
        let Q be the multipath part with the top branching node replaced by a null root;
        for each combination (denoted as ß) of the nodes in the path P do
            generate pattern ß U a with support = minimum support of nodes in ß;
        let freq pattern set(P) be the set of patterns so generated; }
    else let Q be Tree;
    for each item ai in Q do { // Mining multipath FP-tree
        generate pattern ß = ai U a with support = ai .support;
        construct ß's conditional pattern-base and then ß's conditional FP-tree Tree ß;
        if Tree ß ≠ Ø then
            call FP-growth(Tree ß , ß);
        let freq pattern set(Q) be the set of patterns so generated; }
    return(freq pattern set(P) U freq pattern set(Q) U (freq pattern set(P) × freq pattern set(Q)))
}
```

D. *PrePost Algorithm:*

Aim of the proposed algorithm is to maximize the network life by minimizing the total transmission energy using energy efficient routes to transmit the packet. The proposed algorithm is consists of three main steps. The PrePost algorithm was proposed by DENG ZhiHong et. al. in 2012 [6]. This algorithm uses a vertical representation of data called N-List which is derived from an FP-Tree like coding prefix tree called PPC-Tree. The PPC-Tree stores the crucial information about frequent itemsets. Consider the following transaction database [6]:

ID	Items	Ordered frequent itemsets
1	a, c, g, f	c, f, a
2	e, a, c, b	b, c, e, a
3	e, c, b, i	b, c, e
4	b, f, h	b, f
5	b, f, e, c, d	b, c, e, f

The PPC-Tree is constructed from above example:



PPC-tree is a tree structure:

1) It consists of one root labeled as “null”, and a set of item prefix subtrees as the children of the root.

2) Each node in the item prefix subtree consists of five fields: item-name, count, children-list, preorder, and post-order. Item-name registers which item this node represents. Count registers the number of transactions presented by the portion of the path reaching this node. Children-list registers all children of the node. Pre-order is the pre-order rank of the node. Post-order is the post-order rank of the node.

Next, for each node N in a PPC-tree, we call {(N.pre−order, N.post-order): count}

The N-List is thus obtained as follows [6]:

b→ ‹ (4,8):4 ›
c→ ‹ (1,2):1 › - - - ‹ (5,6):3 ›
e→ ‹ (6,5):3 ›
f→ ‹ (2,1):1 › - - - ‹ (8,4):1 › - - - ‹ (9,7):1 ›
a→ ‹ (3,0):1 › - - - ‹ (7,3):1 ›

According the processing sequence, the main steps involved in the PrePost algorithm are:

1) Construct PPC-tree and identify all frequent 1-itemsets;
2) Based on PPC-tree, construct the N-list of each frequent 1-itemset;
3) Scan PPC-tree to find all frequent 2-itemsets;
4) Mine all frequent k (> 2)-itemsets.

Even though the algorithm consumes more memory when the datasets are sparse, it is still the fastest one.

III. EXPERIMENT

In this section we will compare the above mentioned algorithms on the basis of total time required and the total memory usage. There are two main types of datasets i.e. the synthetic data sets and the real data sets.

The synthetic data sets are developed using random generators and the data is not obtained from any real life source. We will be using the dataset "accidents" which is a real data set and it is available at {http://fimi.ua.ac.be/data/}. This data set of traffic accidents is obtained from the National Institute of Statistics for the region of Flanders (Belgium) for the period 1991-2000 [7]. The dataset was donated by Karolien Geurts and contains (anonymized) traffic accident data. In total, 572 different attribute values are represented in the dataset [7].

The test is performed using a system with "Intel(R) Pentium(R) D 3GHz" CPU and "3.21GB" of RAM and the minimum support percentage is set at 50%, 75% and 95%.

Although the dataset contains a total of 3,40,184 traffic accident cases, we will be considering 1,00,000 accident cases for our experiment.

The algorithms were implemented in java programming language and executed in Eclipse IDE.

IV. RESULTS

The below shown line graph shows the Total time required in milliseconds and maximum memory usage in megabytes for Apriori ,ECLAT, FPgrowth and PrePost algorithm respectively. The experiment was carried out for different minimum support which is shown as the X-axis of the graph.

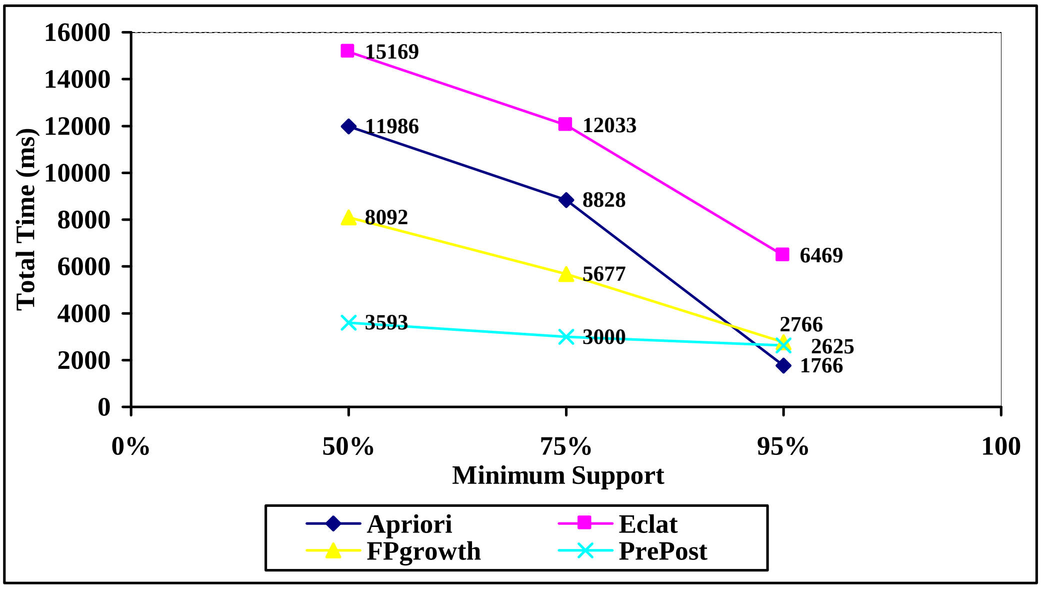


Fig 4.1: Graph for 1,00,000 Accident Cases

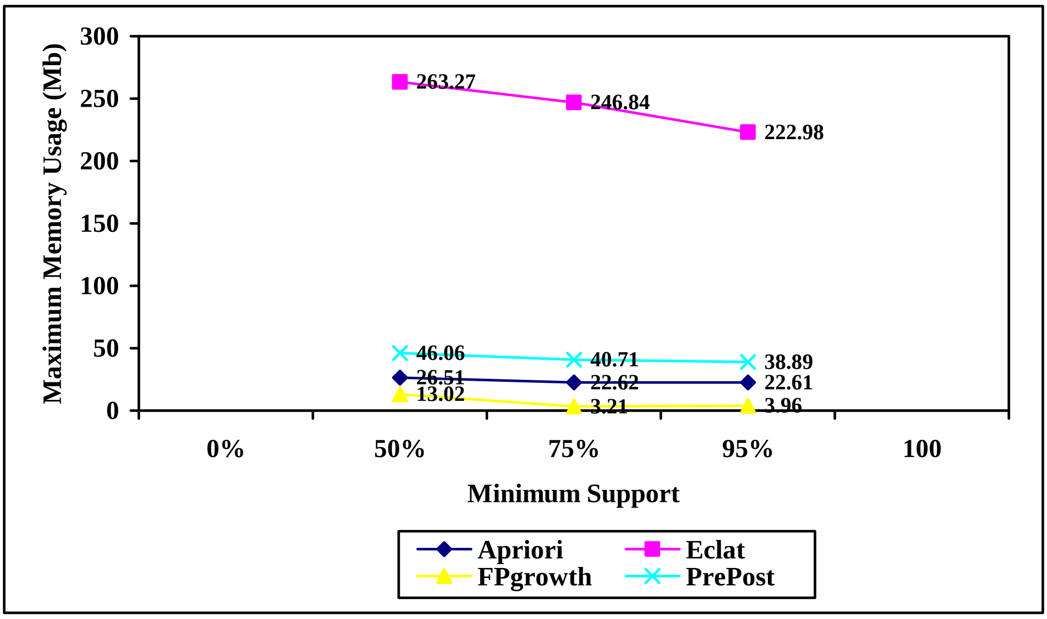


Fig 4.2: Graph for 1,00,000 Accident Cases

V. CONCLUSION

In this paper we surveyed four important frequent itemset mining algorithms namely Apriori, ECLAT, FPgrowth and PrePost algorithm. The major weakness of Apriori algorithm is producing large number of candidate itemsets and large number of database scans. PrePost is the fastest algorithm as it requires least time to complete. But its maximum memory usage is comparatively higher than FPgrowth algorithm. The memory usage is minimum in case of FPgrowth algorithm. Thus the efficiency of PrePost Algorithm can be increased by minimizing the memory usage like FPgrowth Algorithm.

REFERENCES

[1] Rakesh Agrawal and Ramakrishnan Shrikant, 'Fast Algorithms for Mining Association Rules', 20th VLDB Conference, Santiago, Chile, 1994.
[2] Mohammed J. Zaki, 'Scalable Algorithms for Association Mining', IEEE Transactions on Knowledge and Data Engineering, vol. 12, no. 3, May/June 2000.
[3] S.Vijayarani and P.Sathya, 'Mining Frequent Item Sets over Data Streams using ECLAT Algorithm', International Conference on Research Trends in Computer Technologies, 2013.
[4] Manjit Kaur and Urvashi Grag, 'ECLAT Algorithm for Frequent Itemsets Generation', International Journal of Computer Systems (IJCS), Vol. 01-Issue-03, 2014.
[5] Jiawei Han, Jian Pei, and Yiwen Yin, ' Mining Frequent Patterns without Candidate Generation', SIGMOD, 2000.
[6] DENG ZhiHong, WANG ZhongHui and JIANG JiaJian, 'A new algorithm for fast mining frequent itemsets using N-lists, Science China Press and Springer-Verlag, Berlin, Heidelberg, 2012.
[7] Karolien Geurts, 'Traffic Accidents Data Set'," www.luc.ac.be/dam/publications_2003.htm" [Accessed on 1st Feb, 2015], Available: http://fimi.ua.ac.be/data/accidents.dat.