M SUMAN ,T ANURADHA ,K GOWTHAM, A RAMAKRISHNA/ International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 1, Jan-Feb 2012, pp.114-116

A FREQUENT PATTERN MINING ALGORITHM BASED ON FP-TREE STRUCTURE ANDAPRIORI ALGORITHM

M SUMAN[1] , T ANURADHA[2] ,K GOWTHAM[3], A RAMAKRISHNA[4]

1,2ASSOCIATE PROFESSOR
3,4 STUDENTS
1,2,3,4 ECM DEPARTMENT
KLUniversity, Green Fields, Vaddeswaram, Vijayawada, A.P,INDIA

Abstract:

Association rule mining is used to find association relationships among large data sets. Mining frequent patterns is an importantaspect in association rule mining. In this paper, an algorithm named Apriori-Growth based on Apriori algorithm and the FP-tree structure is presented to mine frequent patterns. The advantage of the Apriori-Growth algorithm is that it doesn't need to generate conditional pattern bases and sub- conditional pattern tree recursively.In order to overcome the disadvantages of Apriori algorithm and efficiently mine association rules without generating candidate itemsets,and also the disadvantage of FP-Growth i.e. consumes more memory and performs badly with long pattern data sets. Now in this paper, an algorithm named Apriori-Growth based on Apriori algorithm and FP-Growth algorithm is proposed, this algorithm can combine the advantages of Apriorialgorithm and FP-Growth algorithm.

Key Words: Apriori, datasets, data mining.

1 Introduction

Data mining has recently attracted considerable attention from database practitioners and researchers because it has been applied to many fields such as market strategy, financial forecasts and decision support . Many algorithms have been proposed to obtain useful and invaluable information from huge databases .. Association rule mining has many important applications in our life. An association rule is of the form X => Y. And each rule has two measurements: support and confidence. The association rule mining problem is to find rules that satisfy user-specified minimum support and minimum confidence. It mainly includes two steps: first, find all frequent patterns; second, generate association rules through frequent patterns. Many algorithms for mining association rules from transactions database have been proposed but since Apriori algorithm was first presented. However, most algorithms were based on Apriori algorithm which generated and tested candidate item sets iteratively. This may scan database many times, so the computational cost is high. In order to overcome the disadvantages of Apriori algorithm and efficiently mine association rules without generating candidate itemsets, a frequent pattern tree (FP-Growth) structure is proposed . The FP-Growth was used to compress a database into a tree structure which shows a better performance than Apriori. However, FP-Growth consumes more memory and performs badly with long pattern data sets.. Due to Apriori algorithm and FP-Growth algorithm belong to batch mining. What is more, their minimum support is often predefined, it is very difficult to meet the applications of the real-world. So far, there are few papers that discuss how to combine Apriori algorithm and FPGrowth to mine association rules. In this paper, an algorithm named Apriori-Growth based on Apriori algorithm and FP-Growth algorithm is proposed, this algorithm can efficiently combine the advantages of Apriori algorithm and FP-Growth algorithm. The organization of this paper is as follows. In Section 2, we will briefly review the Apriori method and FP-Growth method. Section 3 proposes an efficient Apriori-Growth algorithm that based on Apriori and the FP-tree structure. Section 4 gives out the conclusions.

2 Two Classical Mining Algorithms

2.1 Apriori Algorithm

Agarwal proposed an algorithm called Apriori to the problem of mining association rules first. Apriori algorithm is a bottom – up , breadth first approach. The frequent item sets are extended one item at a time. It's main idea is to generate k-th

candidate itemsets from the (k-1) –th frequent itemsets and to find the k-th frequent item sets from the k-th candidate item sets. The algorithm terminates when frequent item sets cannot be extended anymore. But it has to generate a large amount of candidate item sets and scans the data as many times as the length of the longest frequent item sets. Apriori algorithm can be written by pseudo code as follows.

Procedure Apriori
Input : Data set D, minimum support minsup
Output: Frequent item sets L

(1) L_1 = find_frequent_1_itemsets(D) ;
(2) For (k=2 ; L_{k-1} ! = ∅ ; k++)
(3) {
(4) C_k=Apriori_gen(L_{k-1} , minsup);
(5) For each transaction t ∈ D
(6) {
(7) C_t =subset(C_k,t);
(8) For each candidate c∈Ct
(9) c.count++;
(10) }
(11) L_k = {c∈C_k | c.count>minsup};
(12) }
(13) Return L = {L_1 U L_2 U L_3 U ……..U L_n};

In the above pseudo code ,Ck means k-th candidate item sets and Lk means k-th frequent item sets.

2.2 FP-Growth Algorithm

Han, Pei et al .proposed a data structure called FP-tree (frequent pattern tree). FP-tree is a highly compact representation of all relevant frequency information in the data set . Every path of FP-tree represents a frequent item set and the nodes in the path are stored
in decreasing order of the frequency of thecorresponding items. A great advantage of FP-tree isthat overlapping itemsets share the same prefix path.So the information of the data set is greatlycompressed. It only needs to scan the data set twiceand no candidate itemsets are required.An FP-tree has a header table. The nodes in theheader table link to the same nodes in its FP-tree.Single items and their counts are stored in the headertable by decreasing order of their counts. Fig.1a showsan example of a data set while Fig.1b shows the FPtreeconstructed by that data set with minsup = 30%.

Transaction
a b c
a c e f
d f
a b c
a c e g
b c

Fig 1a. A data set

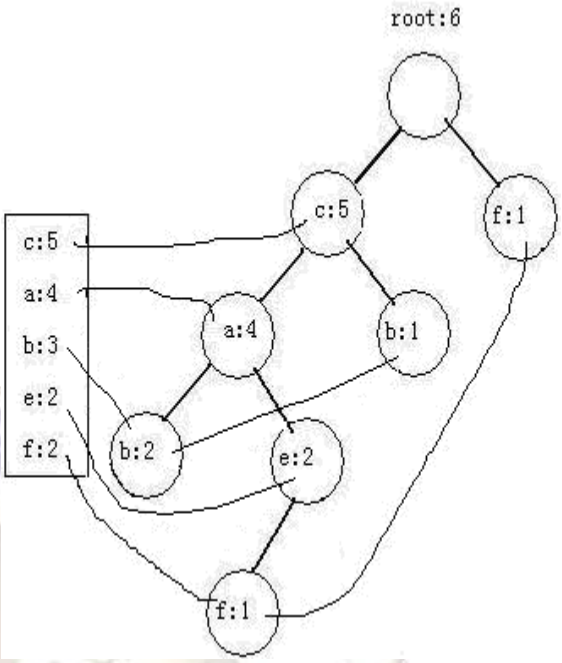

Fig 1b FP-Tree Constructed

Fig 1b. FP-tree constructed by the above datasetThe disadvantage of FP-Growth is that it needs to work out conditional pattern bases and build conditional FP-tree recursively. It performs badly in data sets of long patterns.

3 Apriori-Growth Algorithm

In this section, a new algorithm based on Apriori and the FP-Tree structure is presented , which is called Apriori-Growth .
Fig.2a shows the data structure of the node of header table. Its tablelink points to the first node in FPtree which has the same namewith it. And Fig.2b shows the data structure of the node of FP-tree. Its tablelink points to the next node in FP-tree which has the same name with it.

Name
*tablelink
Count

Fig 2a.The data structure of the node of header table

Name
*tablelink
*parent
*child
count

Fig 2b.The data structure of node of FP-tree

The Apriori-Growth mainly includes two steps. First, the data set is scanned one time to find out the frequent 1 itemsets, and then the data set is scanned again to build an FP-tree as [9] do. At last,

the built FP-tree is mined by Apriori- Growth instead of FP-Growth. The detailed Apriori-Growth algorithm is as follows.

Procedure :Apriori-Growth
Input : data Set D ,minimum support minsup
Output : frequent item sets L

(1) L1= frequent 1 item sets
(2) For(k=2; $L_{k-1} \neq \phi$; k++)
(3) {
(4) Ck= Apriori_Gen(Lk-1 , minsup);
(5) For each candidate c ∈ C_t
(6) {
(7) Sup=FP-treeCalucalate(c);
(8) If(sup >minsup)
(9) $L_k = L_k$ U c;
(10) }
(11) }
(12) Return L = { L_1 U L_2 U L_3 U……U Ln};

Procedure : FP-tree Calculate
Input : candidate item set c
Output : the support of candidate item set c

(1) Sort the items of c by decreasing order of header table;
(2) Find the node p in the header table which has the same name with the first item of c ;
(3) q = p.tablelink;
(4) count = 0;
(5) while q is not null
(6) {
(7) If the items of the itemset c except last item all appear in the prefix path of q
(8) Count + = q.count ;
(9) q = q.tablelink;
(10) }
(11) return count/ totalrecord ;

Function apriori-gen in line 3 generates Ck+1 from Lkin the following two step process:

1. Join step: Generate LK+1, the initial candidates of frequent itemsets of size k + 1 by taking the union of the two frequent itemsets of size k, Pkand Qkthat have the first k−1 elements in common.

LK+1 = Pk∪Qk= {i teml, . . . ,i temk−1, i temk, i temk_ }
Pk= {i teml,i tem2, . . . , i temk−1, i temk}
Qk= {i teml,i tem2, . . . , i temk−1, i temk }
where, i teml<i tem2 <· · · <i temk<i tem_k.
2. Prune step: Check if all the itemsets of size k in Lk+1 are frequent and generate Ck+1 by removing those that do not pass this requirement from Lk+1. This is because any subset of size k of Ck+1 that is not frequent cannot be a subset of a frequent itemset of size k + 1.
Function FP-tree calculate is used for calculating the count of each candidate in the given candidate item set by constructing a FPTree.

4 Future Work

The future work is to improve the Apriori-Growth algorithm such that it determines the frequent patterns for a large database with efficient computation results.

5 Conclusion

In this paper , we have proposed the Apriori-Growth algorithm . This method only scans the dataset twice and builds FP-tree once while it still needs to generate candidate item sets.

6 Acknowledgements

This work was supported by Mr. M Suman Assoc. Professor, and Mrs T AnuradhaAssoc. Professor Department of Electronics and Computer Science Engineering, KLUniversity.

7 References

[1] Agrawal, R et.al.,(1994) Fast algorithms for mining association rules.In Proc. 20th Int. Conf. Very Large Data Bases.

[2] Agrawal,R et al.,(1996), “Fast Discovery of Association Rules,” Advances in Knowledge Discovery and Data Mining.

[3]. Agrawal, R., Imielinski, T., and Swami et. al (1993). Mining association rules between sets of items in large databases. In Proceedings of the ACM SIGMOD International Conference on Management of Data, 207-216.

[4] M.S. Chen, J. Han, P.S. Yu, “Data mining: an overview from a database perspective”, *IEEE Transactions on Knowledge and Data Engineering*, 1996, 8, pp. 866-883.

[5] J. Han, M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publisher, San Francisco, CA, USA, 2001.