Hybrid Association Rule Mining using AC Tree

Preethi Kolluru Ramanaiah[1*]

1. Business Technology Analyst, Deloitte, Bangalore, India

* E-mail: preethiram4@gmail.com

Abstract

In recent years, discovery of association rules among item sets in large database became popular. It gains its attention on research areas. Several association rule mining algorithms were developed for mining frequent item set. In this papers, a new hybrid algorithm for mining multilevel association rules called AC Tree i.e., AprioriCOFI tree was developed. This algorithm helps in mining association rules at multiple concept levels. The proposed algorithm works faster compared to traditional association rule mining algorithm and it is efficient in mining rules from large text documents.

Keywords: Association rules, Apriori, FP tree, COFI tree, Concept hierarchy.

1. Introduction

At present, association analysis [8] was widely used in many industries. This has significant usage in market-basket analysis. The main aim of association analysis is the discovery of association rules of the form attribute-value. The 2 major sub problems in mining association rules are:

- Finding all frequent item sets that appear more than minimum support a
- Generating association rules using these frequent item sets.

A number of algorithms for mining frequent item sets have been proposed after Agarwals [7] first work of deriving categorical association rules from transactional databases. And then, many new algorithms have been developed for efficient mining. Now, these existing algorithms can be categorized in two classes called candidate-generate approach and pattern growth approach. The best suited example for candidate generate approach is Apriori[6] and pattern growth approach is Fp Tree[7].

1.1 Candidate-generate Approach

The candidate-generate approach is most commonly used and most widely popular approach for mining data from transactional databases. The two main steps in this method are join and prune[8,7]. Here k-frequent itemsets are joined to form (k+1) th candidate itemsets. In each and every iterations, the database is scanned o discard itemsets that do not meet minimum support. This algorithm is called as Apriori algorithm. It achieves good reduction on the size of candidate sets. However, it takes many scans of database to check the candidates support and this suffer from larger overhead of I/O scans. This is consider to be the main disadvantage of Apriori algorithm on mining large text documents.

1.2 Pattern growth Approach

The next set of algorithms for mining association rules are pattern growth approach [6]. Over the past few years, many pattern-growth approaches have been developed. Some of the most popular algorithms are Fp Growth, Tree-projection, H-mine, COFI, BOMO, etc. The main aim of pattern-growth approach is the use of tree to store database instead of generating candidates. Thus this method proves to be much faster compared to candidate-generate approach. However, it suffers from the problem of memory overhead for mining data from large database. But for smaller data set, this approach is consider to be most suited approach.

2. Multilevel Association Rule Mining

Important development in pattern-growth approach is multilevel association rule mining which involves items at different levels of abstraction. From [7], this method involves 2 main stage

- construction of a modified frequent pattern tree
- Repetitive building of small data structures.

Here, association rules are generated at multiple level using the frequent patterns at related concept level. The best suited example for multilevel association rule mining algorithm is COFI tree called as Co-Occurrence Frequent Item tree.

2.1 COFI tree

COFI tree[3, 4] is mainly used for finding more interesting patterns hidden at multiple concept level. It deals with the construction of relatively small trees for each frequent item in the header table of Fp tree. Thus the main important step in COFI tree first involves construction of Fp header table. Then pruning is done to remove all non-frequent itemsets with respect to minimum support threshold. Then at each level, it eliminates all non-frequent itemsets that will not participate in subsequent levels. This is similar to Apriori property [5]. The elimination at each of the levels is to find all frequent patterns with respect to one frequent item. The process is carried out for each and every element in the header table of Fp tree. Subsequently, the join process is carried out at every level to generate candidate like itemset for performing next iteration. Since, it requires the construction of Fp tree for each and every frequent item in the header table, the memory requirement is more. Hence it is not scalable for larger dataset.

3. Hybrid AC Tree Algorithm

In order to efficiently mine frequent patterns from larger data set, the new algorithm called AC tree is developed. It is hybrid algorithm that combines Apriori and COFI tree and FP tree. The main aim of the project is to overcome the disadvantage of those traditional algorithms and make it suitable for mining larger dataset efficiently. Figure 3.1 shows the complete architecture.

3.1 AC tree

The AC tree mainly involves 2 steps. In the first step, the dataset is scanned one time to find out frequent 1-itemsets using Apriori algorithm. The main aim of this step is to increase the efficiency in doing text mining. The text document will have many words which are irrelevant for analysis. Thus as a initial step, data preprocessing is carried out to remove all stop words. Then, using Apriori algorithm, 1- frequent itemset is discovered. The idea behind discovering 1 frequent itemset and not more is, this is enough to give sufficient result which helps to carry out further steps. In the second step, Fp header table is constructed using those 1- frequent itemset. For each and every item in the Fp header table, a tree is constructed which contains all the transactions that particular element is presented. Only considering those transactions where particular element , say x, is presented. Fp tree is constructed for element x. Likewise, the process repeated for each and every element. Then frequent patterns are found out at every level of iteration. Pruning is done based on its minimum support count. Thus frequent pattern is mined at each and every level hence it is called as multilevel association rule mining algorithm. Since it is multilevel rule mining algorithm, the number of iterations depends on the type of dataset involved. The mining process is recursive until there exist 1 frequent itemset at i^{th} level. This is based on assumption that, if number of patterns in nth level is 1, then there exist no frequent itemset in (n+1)th level. The algorithm is as follows.

Algorithm

Step 1: Scan the data set once In this process the given text dataset is scanned fully and the word frequency is noted for each and every word

Step 2: generate all 1-frequent item sets that satisfy minimum support count using Apriori

Step 3: sort the items in candidate item set in decreasing order of frequency

step 4: now choose the lowest frequent item set 'x' and make note of all transactions that contains the item x

step 5: construct a Fp tree for each and every item set x along with the transactions that contain item x

step 6: find the frequent patterns at each level of construction of Fp tree

step 7: repeat step 4 until sub conditional Fp tree is constructed for each and every element step 8: now perform join operation of all conditional sub Fp trees

step 9: at each and every level of join operations, find the frequent item sets.

Thus, from the above algorithm, we mine frequent items at multiple levels.

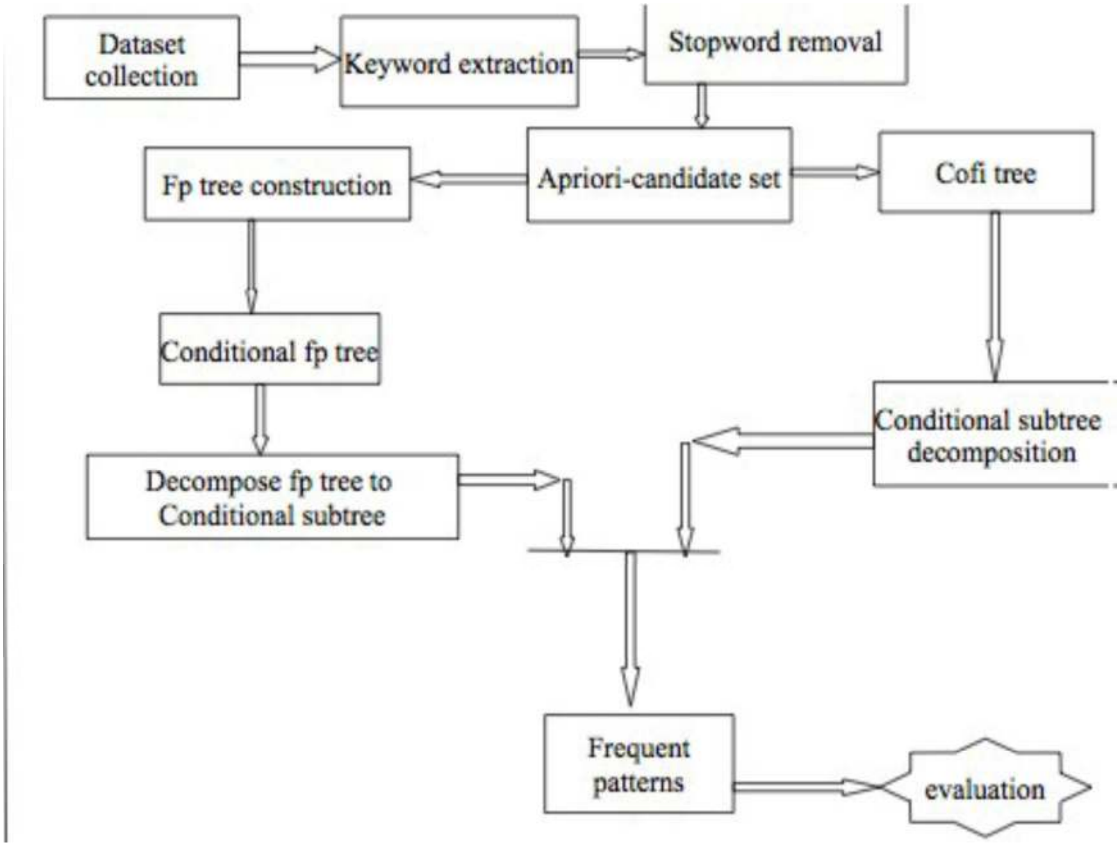

Figure 3.1 Architecture of AC tree

4.Experimental Results

The proposed algorithm was implement on java. The dataset used for analysis is 20 Newsgroup dataset. In order to verify its performance, the AC tree algorithm was compared with Apriori , Fp tree and APFT[8] algorithm. Experiments shows that AC tree outperforms the above traditional algorithms. It works faster and helps to discover more interesting patterns at multiple level. This algorithm is efficiently used for mining a large corpus text documents.

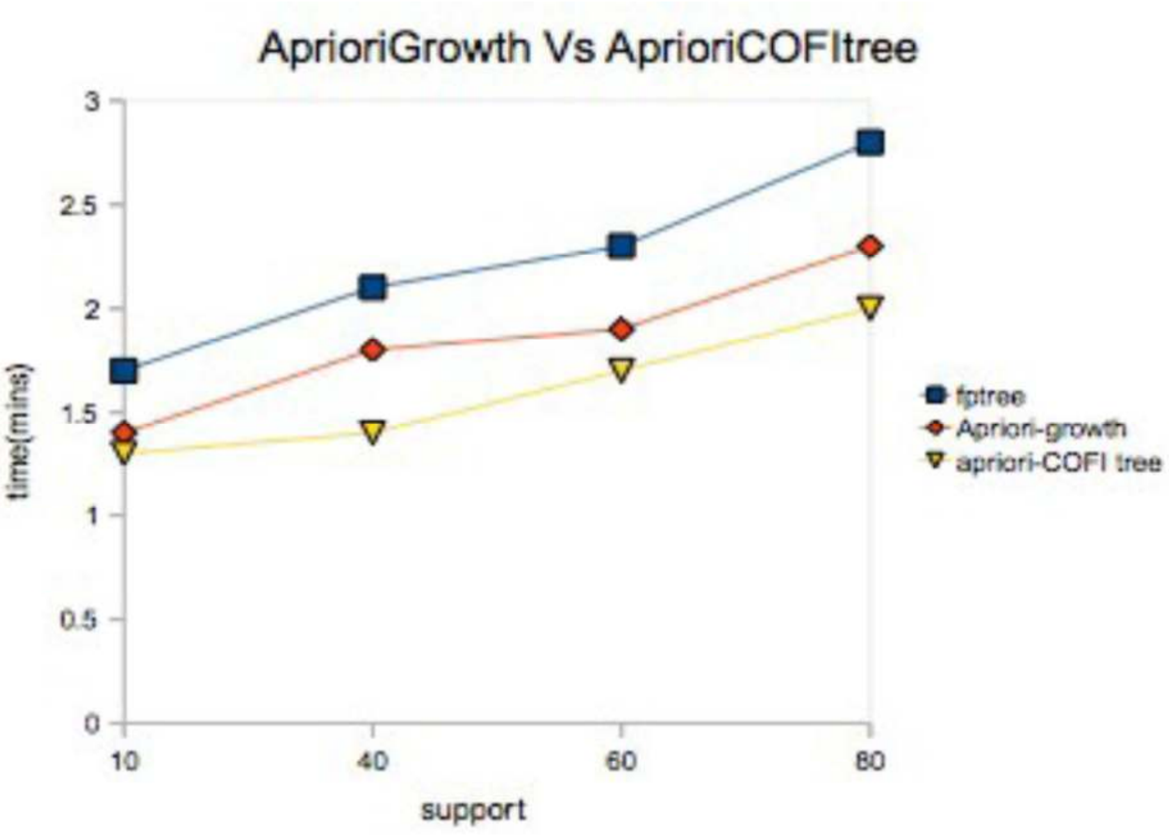

Figure 4.1 Comparisons between AprioriGrowth and AprioriCOFI(AC Tree)

The above chart i.e. Figure 4.1 shows the comparison between AprioriGrowth, AprioriCOFI (AC) tree and Fp tree. In order to find all nth level item sets, AC tree yields a better result. And also time taken for mining frequent item sets are comparatively minimum in AC tree than Apriori growth algorithm. The number of patterns mined is more interesting because it helps to identify the hidden interesting patterns at multiple levels. Thus AC tree works better than Apriori growth and Fp tree.

References

[1] Qihua Lan, Defu Zhang, (2009) “A new algorithm for frequent itemsets mining based on Apriori and FP-tree”, Ref no. 978-0-7695-3571-5/09 2009 IEEE, global congress on Intelligent systems, pp. 1099-1102.

[2] Mohammad el-hajj, Omar R.Zaiane, (2005) “COFI Tree mining: a new approach to pattern growth with reduced candidacy generation” International Journal of Business Intelligence and Data Mining Volume 1 Issue 1, July 2005, pp. 88-106.

[3] Sze-chung Ngan, Tsang Lam, Raymond chi-wing wong and Ada wait-chee fu (2005) “ mining N most interesting itemsets without support threshold by COFI -tree” , Int. J.Business Intelligence and Data mining, vol. 1, No. 1, 2005, pp. 348-354.

[4] Mohammad El-Haii and Osmar R, (2003) “COFI-tree mining: A new approach to pattern Growth within the context of interactive mining” In Proc. 2003 Int'1 Conf. On Data mining and Knowledge Discoverey (ACM SIGKDD), August 2003 pp. 109-118

[5] Yongwppk yoon and Gary geunbae lee, (2003) “practical application of associative classifier for document classification”, BK21 program of Korea Ministry of education and grant number R01-2003-000-10181-0, pp. 331-339.

[6] J.Han, J.Pei and Y.Yin.(2000), "Mining frequent patterns without candidate Generation", Proceeding of ACM SIGMOD International Conference Management of Data, 2000, pp. 1-12.

[7] Agrawal R and R.Srikant, (1994) "fast algorithms for mining association rules", In VLDBY94, ISBN:1-55860-523-1, pp.487-499.

[8] Agrawal R, T.Imielinski and A.Swami (1993). "Mining association rules between set of items in large databases" In Proc.1993 ACM-SIGMOD Int conf. Management of Data, Washington.D.C, may 1993, pp. 207-216.