



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

Mining of Web Server Logs using Extended Apriori Algorithm

SANJEEV DHAWAN¹, SWATI GOEL²

¹ Faculty of Computer Science & Engineering, University Institute of Engineering and Technology (U.I.E.T.), Kurukshetra University, Kurukshetra-136119, Haryana, INDIA

² M.Tech. Research Scholar, University Institute of Engineering and Technology (U.I.E.T.), Kurukshetra University, Kurukshetra-136119, Haryana, INDIA

Abstract: Association rule mining is one of the most significant techniques in the field of data mining. It is very useful in discovering relationships hidden in large transaction datasets such as frequent patterns, associations etc. One of the popular and important algorithms in this category is Apriori algorithm which finds frequent itemsets using an iterative approach. But it suffers from a major limitation that in case of large databases, it requires a large number of passes while searching the frequent itemsets, thus increasing its scanning time. In order to lessen this time, an improved version of Apriori algorithm, called Extended Apriori is proposed in this paper which decreases the number of transactions in the database, hence reducing size of the database so as to minimize the scanning time. This extended algorithm is then used to mine web server logs of an educational web site in order to discover frequently visited pages by the user and also its performance is compared graphically with existing Apriori algorithm. In the end, the paper also outlines some future research directions in the area of web server log mining.

Keywords: Data mining, Web Usage mining, Association rule, Apriori algorithm, Frequent patterns

I. Introduction

Data mining may be defined as extracting knowledge from a large amount of data by the application of various techniques such as statistical methods, classification, clustering, association rules etc. [1]. Association rule mining is one of the important techniques used in data mining which is used to find associations among a group of frequently accessed items, with support exceeding a pre specified threshold. Support of an itemset may be defined as the percentage of transactions containing the itemset. A popular example of this technique is market basket analysis which analyses buying habits of customers by finding associations between items in their baskets [1]. For example, if computer and speakers appear frequently together in a transaction database then it is a frequent itemset. Association rules are also used in other areas such as e-commerce, marketing and promotional campaigns, website optimization and personalization etc.

It is evident from the literature that frequent pattern mining is a very expensive task as counting the occurrence of itemsets in database takes a large amount of computational time. As a result, many algorithms were proposed to improve its processing time. One such algorithm is Apriori which is a type of association rule mining algorithm which finds frequent itemsets using a bottom up approach [9]. But its main limitation is that it takes a large amount of time in scanning large datasets. Also, the numbers of candidate itemsets generated are huge. To remove these limitations, many improved variations of apriori have been proposed. The proposed improved Apriori algorithm called Extended Apriori removes the limitation of large scanning time by reducing the size of database which is done by removing unnecessary transactions.

This rest of paper is divided into following sections: section 2.0 describes the related work, section 3.0 covers the existing and proposed algorithm, Section 4.0 uses proposes algorithm to mine the patterns from web server logs of an educational web site in order to discover frequently visited pages by a user and also shows graphically, the comparison results between the existing and proposed algorithm using three parameters: support count, transaction count and running time. Section 5.0 concludes the paper and outlines some of the promising areas of the future research.

II. Related work

Apriori algorithm used for frequent pattern mining was proposed by Aggarwal *et al.* [7]. Though many variations of this algorithm exist till date, but Apriori is still an area which needs further research. Many variations done on the Apriori algorithm is presented in this section:

The AIS (Agrawal, Imielinski, Swami) algorithm [6] was the first algorithm to address the problem of generating association rules. It uses candidate itemsets in order to find frequent itemsets. It produces candidate itemsets on the fly on each scan of the database and compares them with already found itemsets to check if they are present in the previous dataset. The disadvantage of this algorithm was that it generates too many candidate itemsets and also requires a large number of database scans, hence increasing the processing time of the algorithm.

Aggarwal *et al.* [7] also presented various versions of Apriori namely: Apriori, AprioriTid and AprioriHybrid. Apriori algorithm firstly finds those itemsets that are likely to be frequent. These are known as candidate itemsets. These are generated by joining the previously found frequent patterns. Apriori property is used to prune unnecessary patterns from candidate itemsets, thus reducing its size. Next, it finds those itemsets that are actually frequent by using a minimum support. AprioriTID uses the same candidate generating technique of Apriori with a difference that it does not use the same database to search for the frequent itemsets. Rather it constructs a database by encoding the candidate itemsets which is smaller in size than the original database. Its main advantage is that it is faster than Apriori algorithm. A further improvement in this field was done with AprioriHybrid, which uses Apriori in the initial stages but shifts to AprioriTID in the later stages. A Direct Hashing and Pruning (DHP) algorithm proposed by [2] generates candidate itemsets using a hash based technique. It removes those entries from hash table which does not fulfill minimum support criteria, thus improving its efficiency. It also utilizes a pruning technique to reduce the database size. Equivalence Class Clustering and Bottom-up Lattice Traversal (Eclat) [5] uses a depth first search technique in contrast to Apriori which uses a bottom up mechanism. It partitions the candidates into equivalence classes which are disjoint to each other. The local database is scanned only once.

FP- Growth was an effort to improve Apriori by introducing a data structure called frequent pattern tree. Han *et al.* [3] proposed an algorithm called FP-growth for frequent pattern mining. It uses a divide and conquer mechanism and does not generate candidate itemsets. Though many algorithms exist for association mining it still suffers from limitation of scanning the database multiple times. Therefore, it is needed to address the issues of reducing the time of database scan. This limitation and some other related problems made us to continue the research work in this particular area. In this paper an improved version of Apriori algorithm called Extended Apriori algorithm is proposed which reduces the database size so as to reduce the time of scanning the database.

III. Existing and Proposed algorithm

A. Existing algorithm: Apriori

Apriori is an algorithm proposed by R. Agrawal *et al.* [7] in 1994 for mining frequent itemsets. Apriori uses an iterative bottom up approach, where k-itemsets are used to find (k+1)-itemsets. The procedure is divided into following two steps:

- a. Join step: Set of frequent 1-itemsets is searched by scanning the database to determine the count for each itemset, and considering only those items that fulfill minimum support criteria. This set is represented as L1. Support of an itemset is the percentage of database transactions in which it appears. In general terms, L_{k-1} is joined with itself in order to find candidate k-itemsets denoted as C_k, which is then used to find L_k in the prune step by using the minimum support criteria [4].
- b. The prune step: Database is scanned to find the count of each candidate in C_k which will find L_k. C_k can be large, and so this could require heavy processing. To reduce its size, the Apriori property is used in which if any k-1 subset of a candidate k-itemset does not lie in L_{k-1}, then the candidate itemset can be removed from C_k. Only those itemsets are included in L_k that satisfy the minimum support count criteria [4].

1. Improving the efficiency of Apriori

- Transaction reduction: It reduces the number of transactions in the database. One of the methods is to remove all those transactions that do not contain frequent k-itemsets while searching for frequent k+1 itemsets.
- Reducing the number of passes over data: Many techniques exist for achieving this. One of the methods is to use partitioning method [8]. A partitioning technique needs only two database scans to mine the frequent itemsets.
- Reducing the number of candidates generated.

B. Proposed algorithm: Extended Apriori algorithm

The traditional Apriori algorithm suffers from two major bottlenecks: large amount of time is required in scanning database and large number of candidate itemsets generation. Proposed algorithm addresses the first problem by

reducing the time for scanning database which is done by reducing the number of database transactions by incorporating the following two features:

- (i) Before determining the count of candidate k-itemsets from the database, only those transactions are considered in which length of transaction is greater than or equal to k while those having length less than k are eliminated.
- (ii) A number field is added along with each database transaction. For each transaction it is tested whether a subset of candidate itemset is present in a transaction and for those transactions it is true, the number field is incremented by one. This process is repeated for each candidate itemset. After the count has been accumulated, those transactions are removed that have $Number < (2^{\text{length of transaction}} - 1)/2$, where $(2^{\text{length of transaction}} - 1)$ denotes the total number of subsets of itemsets in a transaction.

The left over transactions are taken as input for pruning stage. The algorithmic steps in the form of Pseudo code are shown below:

1. Algorithm:

Input:

D: a database containing transactions;

min_supp: the minimum support threshold.

Output:

O: frequent itemsets in database, *D*.

Method:

Scan *D* to generate *C*₁ and use it to find *L*₁;

For (*k* = 2; *Lk*-1 ≠ 0; *k*++) do begin

Ck = apriori gen(*Lk*-1) // generates candidates and uses Apriori property

For each transaction *t* ∈ *D* do begin

If(*t*.length < *k*)

Delete *t*;

End

For each candidate *c* ∈ *Ck* do begin

For each transaction *t* ∈ *D* do begin // scan *D* for subset count

Ct = subset(*Ck*, *t*); // get the subsets of *t* that are candidates

t.number++

End

End

For each transactions *t* ∈ *D* do Begin

If((*t*.number < $2^{\text{length of } t} - 1$)/2)

Delete *t*

End

For each transaction *t* ∈ *D* do begin

Ct = subset(*Ck*, *t*); // get the subsets of *t* that are candidates

For each candidate *c* ∈ *Ck* do begin // scan *D* for subset count

c.count++

End

End

Lk = { *c* ∈ *Ck* | *c*.count ≥ *min_supp* };

End

2. Description of Pseudo code [4]:

1. Frequent 1-itemsets, *L*₁ are found. Each item is taken in candidate 1-itemsets, *C*₁. The algorithm then scans all of the transactions to count the number of times each item occurs. The set of frequent 1-itemsets, *L*₁ is then computed which consists of the candidate 1-itemsets satisfying minimum support count.
2. The apriori gen procedure [7] generates the candidates and then utilizes Apriori property to remove those itemsets that have an infrequent subset.
3. Then, it is checked if the length of transaction is less than the number of the current iteration. If it returns true, then that corresponding transaction is deleted because those transactions would serve no purpose during scanning.

4. In the next step, database is checked to see how many candidate itemsets are contained in each transaction. If the accumulated count is less than half the number of subsets of each transaction, then that corresponding transaction is deleted.
5. Once the unnecessary transactions are deleted, a subset function is applied to each transaction, in order to search for all subsets of transaction which are candidates and the corresponding counts for each of the candidates are collected.
6. Finally, all those candidates that satisfy minimum support count, constitute the set of frequent itemsets, L.

IV. Mining Web Server logs using Extended Apriori algorithm

In this work, web log files of the month March 2013 is taken from the web site www.uietkuk.org. This is an engineering college web site and provides information of engineering and PG subjects, faculty details, syllabus, courses offered, etc. This knowledge mined from this web site can be used for web personalization, web site design modification, discovering frequently visited pages by the user etc. Both the data preprocessing steps namely, data cleaning, session identification and user identification and pattern discovery was done by using language JAVA SDK 6.0, Netbeans IDE and Windows XP operating system. Performance comparison with the help of graphs is shown in Figures. 1 and 2 which indicates that Extended Apriori algorithm is more efficient as it takes lesser time in both cases. The major advantage of this approach is that the algorithm results in reduced time in database scans as size of database is greatly reduced. To make the comparison fair, algorithms are implemented using the same data structures in java namely, hash map, array list and Properties, each having its distinct properties. Test data set of same is used having same number of transactions for generating the graph in figure 1 whereas test data sets of different sizes are used having different number of transactions for generating the graph in figure 2. Y-axis of both graphs refers to the running time in seconds and X-axis refers to the minimum support constraint in fig 1 and to the number of transactions in fig 2. Figure 1 shows that, as the support count increases, running time of the algorithm decreases. Figure 2 shows that, as number of transactions increases, running time also increases.

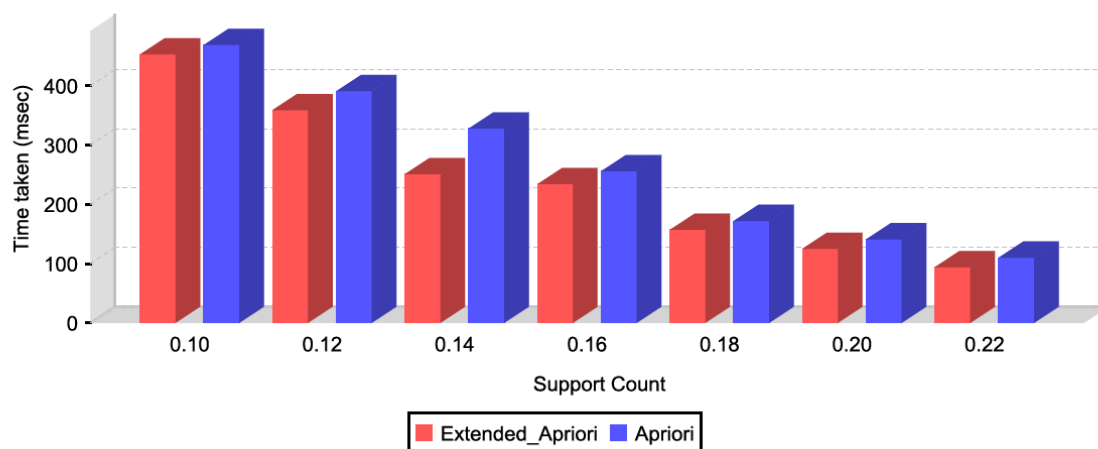


Figure 1 Plot of Support count vs. Time taken

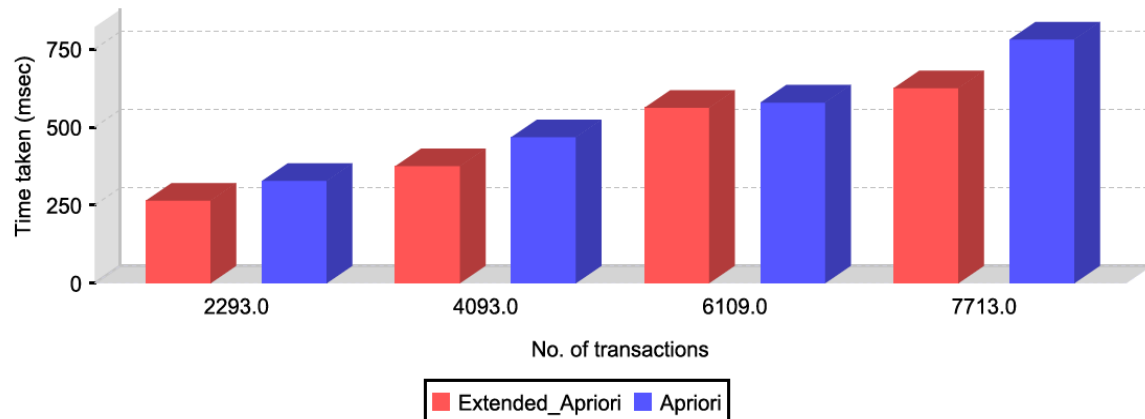


Figure 2 Plot of No. of transactions vs. Time taken

V. Conclusions and future research directions

In this paper, the improved version of existing Apriori algorithm known as Extended Apriori algorithm is proposed to overcome the limitations of the basic Apriori algorithm. The new proposed method generates smaller database which reduces the time of database scans. The improved version Apriori algorithm is more efficient which takes less time and hence reflects high efficiency. Present work is on association rule mining but future research will be done using other data mining techniques such as classification, clustering etc. on web server logs to extract meaningful patterns. Also, the proposed method will be compared with the other existing algorithms based on time and space complexity and other tools such as MATLAB will be adopted for its simulation.

References

- [1] G.K. Gupta, "Introduction to data mining with case studies, Second edition", 2006.
- [2] J.S. Park, M.chen, P.S. Yu, "An effective hash based algorithm for mining association rules", ACM SIGMOD International Conference on Management of Data, May 1995
- [3] Jiawei Han, Jian Pei, Yiyen Yin, "Mining Frequent Patterns without Candidate Generation", Proceedings of the ACM SIGMOD International Conference on Management of Data, Dallas, Texas, USA, May 16-18, 2000.
- [4] Jiawei Hwan and Micheline Kamber, "Data Mining: Concepts and techniques, Second edition", April 012.
- [5] Mohammed Javeed Zaki, Srinivasan Parthasarathy, Mitsunori Ogihara, Wei Li, "New Algorithms for Fast Discovery of Association Rules", The 3rd International Conference on Knowledge Discovery and Data Mining, 1997.
- [6] Rakesh Agrawal, Tomasz Imielinski, Arun Swami, "Mining association rules between sets of items in large databases", Proceedings of the ACM SIGMOD Conference on Management of Data, Washington, D.C., 1993.
- [7] Rakesh Agrawal, Ramakrishnan Srikant, "Fast algorithms for mining association rules", Proceedings of 20th International Conference on Very Large Data Bases, Chile, September, 1994.
- [8] S.Veeramalai, N.Jaisankar and A.Kanna, "Efficient Web Log Mining Using Enhanced Apriori Algorithm with Hash Tree and Fuzzy", International journal of computer science & information Technology (IJCSIT), Vol.2, No.4, August 2010.
- [9] Suneetha K R and Krishnamoorti R, "Web Log Mining using Improved Version of Apriori Algorithm", International Journal of Computer Applications, Volume 29– No.6, September 2011.