International Journal of Computer Applications (0975 – 8887)
Volume 145 – No.2, July 2016

Comparative Study and Analysis on Frequent Itemset Generation Algorithms

Aasma Parveen
M.E Scholar
Department of Computer Science and Engineering
Faculty of Engineering and Technology
Shri Shankaracharya Technical Campus, Junwani, Bhilai, District-Durg, Chhattisgarh-490020, India

Shrikant Tiwari
Assistant Professor
Department of Computer Science and Engineering
Faculty of Engineering and Technology
Shri Shankaracharya TechnicalCampus, Junwani, Bhilai, District-Durg, Chhattisgarh-490020,India

ABSTRACT
Association mining aspire to extort frequent patterns, interesting correlations, associations or informal structures between the sets of items in the transaction databases or further data repositories. It plays a essential role in spawning frequent item sets from big transaction databases. The finding of interesting association relationship between business transaction records in various business decision making process such as catalog decision, cross-marketing, and loss-leader analysis. It is also utilized to extort hidden knowledge from big datasets. The Association Rule Mining algorithms such as Apriori, FP-Growth needs repeated scans over the whole database. All the input/output overheads that are being generated through repeated scanning the whole database reduce the performance of CPU, memory and I/O overheads. In this paper we have equaled many classical Association Rule Mining algorithms and topical algorithms**.**

Keywords
Data Mining, Association Rule Mining (ARM), Association rules, Apriori algorithm, Frequent pattern.

1. INTRODUCTION
The quick development of computer technology, specially enhance capacities and reduce costs of storage media, has led businesses to accumulate large amounts of external and internal information in big databases at low cost. Mining helpful information and useful knowledge from these big databases has thus evolved into a significant research area [3, 2, 1].

Association rule mining (ARM) [18] has become one of the core data mining tasks and has concerned tremendous interest amongst data mining researchers. ARM is an un-directed or un-supervised data mining method which works on variable length data, and generates clear and understandable results. Association Rule Mining (ARM) algorithms [17] are defines into two category; namely, algorithms respectively with applicant generation and algorithms with no applicant generation. In the initial category, those algorithms which are similar to Apriori algorithm for applicant generation are considered. Eclat may also be considered in the initial category [8]. In the second category, the FP-Growth algorithm is the best known algorithm.

The major disadvantage of earlier algorithms is the repetitive scans over big database. This may be a reason of decrement in CPU performance, memory and increase in I/O overheads. The performance and efficiency of ARM algorithms mostly depend on three factors; namely applicant sets generated, data structure utilized and details of implementations [8]. ARM is an un-directed or un-supervised data mining method which works on variable length data, and generates clear and understandable results. Association Rule Mining (ARM) algorithms [17] are defined into two categories; namely, algorithms respectively with applicant generation and algorithms with no applicant generation. In the initial category, those algorithms which are similar to Apriori algorithm for applicant generation are considered. Eclat may also be considered in the initial category [8]. In the second category, the FP-Growth algorithm is the best known algorithm. Following table defines the comparison among these three algorithms [9].

Algorithm	Scan	Data Structures
Apriori	M+1	HashTable &Tree
Eclat	M+1	HashTable &Tree
FP-Growth	2	PrefixTree

The major disadvantage of above discussed algorithms is the repetitive scans of large database. This may be a cause of decrement in CPU performance, memory and increase in I/O overheads. The performance and efficiency of ARM algorithms mostly depend on three factors; namely applicant sets generated, data structure utilized and details of implementations [8].

The remainder of this paper is organized as follow: Section 2 provides a short review of the related work. In Section 3 we describe Frequent Item set and Association Rule Mining through Apriori Algorithm. In Section 4, we have explained the problem in topical algorithm and how efficiency of similar algorithm can be measured and how speed up is decided. In section 5 we have concluded our study.

2. RELATED WORK
One of the mainly well-known and popular data mining methods is the Association rules or frequent item sets mining algorithm. The algorithm was originally proposed by Agrawal et al. [4] [5] for market basket investigation. Because of its important applicability, various revised algorithms have been introduced since then, and Association rule mining is still a broadly researched area.

Agrawal et al. presented an AIS algorithm in [4] which generate applicant item sets on-the-fly during every pass of the database scan. Large item sets from earlier pass are checked if they are present in the current transaction. Thus new item sets are formed by extending obtainable item sets. This algorithm turns out to be useless because it generates too many applicant item sets. It needs more space and at the same time this algorithm needs too many passes over the entire database and also it generates rules with one consequential item.

Agrawal [5] developed several versions of Apriori algorithm such as Apriori, AprioriTid, and AprioriHybrid. Apriori and AprioriTid produce item sets utilizing the big item sets found in the previous pass, without considering the transactions. AprioriTid progress Apriori by utilizing the database at the initial pass. Counting in consequent passes is done using encodings created in the initial pass, which is much smaller than the database. This lead to a dramatic performance development of three times faster than AIS.

Scalability is another significant area of data mining because of its large size. Hence, algorithms must be capable to "scale up" to handle big amount of data. Eui-Hong et al. [16] try to create data distribution and applicant distribution scalable by Intelligent Data Distribution (IDD) algorithm and Hybrid Distribution (HD) algorithm respectively. IDD addresses the problems of communication overhead and unnecessary computation by using aggregate memory to partition applicants and move data efficiently. HD progress over IDD by dynamically partitioning the applicant set to maintain good load balance. Different works are reported in the literature to modify the Apriori logic so as to progress the efficiency of generating rules. These systems even though focused on reducing time and space, in real time still requires improvement.

3. FREQUENT ITEM SET AND ASSOCIATION RULE

The objective of Association rule mining is exploring relations and essential rules in large datasets. A dataset is considered as a series of entries consisting of attribute values also called as items. A set of such item sets is known an item set. Frequent item sets are sets of pages which are visited frequently mutually in a single server session.

Let I ={ I1, I2, … , Im }be a set of items. Let D, the task-relevant data, is a set of database transactions where every transaction T is a set of items such that $T \subseteq I$. Every transaction is associated with an identifier, known as TID. Let A be a set of items. A transaction T is said to contain A if and only if $A \subseteq T$. An association rule is an insinuation of the form $A \Rightarrow B$, where $A \subset I$, $B \subset I$, and $A \cap B = \emptyset$. The rule $A \Rightarrow B$ hold in the transaction set D with support s, where s is the percentage of transactions in D that contain $A \cup B$ (i.e., the union of sets A and B, or say, both A and B). This is taken to be the possibility, $P(A \cup B)$. The rule $A \Rightarrow B$ has confidence c in the transaction set D, where c is the percentage of transaction in D containing A that also contain B. This is taken to be the conditional possibility, $P(B|A)$. That is,

support $(A \Rightarrow B) = P(A \cup B)$……………..(2.1)

confidence $(A \Rightarrow B) = P(B|A)$………….(2.2)

A set of items is refers to as an item set. An item set that contains k items is a k-item set. The set {bread, butter} is a 2-item set. The occurrence frequency of an item set is the number of transactions that contains the item set; it is also called as the frequency, or support count. If the virtual support of an item set I satisfy a pre specified least support threshold then I is a frequent item set. The set of frequent k-item sets is normally denoted by Lk. From Equation (2.2), we have

$$\text{confidence } (A \Rightarrow B) = P(B|A) = \frac{\text{support } (A \cup B)}{\text{support } (A)} = \frac{\text{support\_count } (A \cup B)}{\text{support\_count } (A)}$$

Let τ = I1, I2... Im be a set of binary attributes, called items. Let T be a database of transactions. Each transaction t is represented as a binary vector, with t[k] = 1 if t bought the item Ik, and t[k] = 0 otherwise. There is one tuple in the database for each transaction. Let X be a set of some items in τ. We say that a transaction t satisfies X if for all items Ik in X, t[k] = 1.

By an association rule, we mean an implication of the form $X \Rightarrow Ij$, where X is a set of some items in τ, and Ij is a single item in τ that is not present in X. The rule $X \Rightarrow Ij$ is satisfied in the set of transactions T with the confidence factor $0 \le c \le 1$ if at least c% of transactions in T that satisfy X also satisfy Ij. We will use the notation $X \Rightarrow Ij \mid c$ to specify that the rule $X \Rightarrow Ij$ has a confidence factor of c.[3]

3.1 Apriori Algorithm

The Apriori algorithm is one of the most famous algorithms for mining frequent patterns and association rules [4]. It introduces a system to generate applicant item sets Ck in the pass k of a transaction database utilizing only frequent item set Lk−1 in the earlier pass. The idea rests on the fact that anysubset of a frequent item set must be frequent as well. Hence, Ck can be created by joining two item sets in Lk−1 and pruning those that contain any subset that is not frequent as shown in Figure

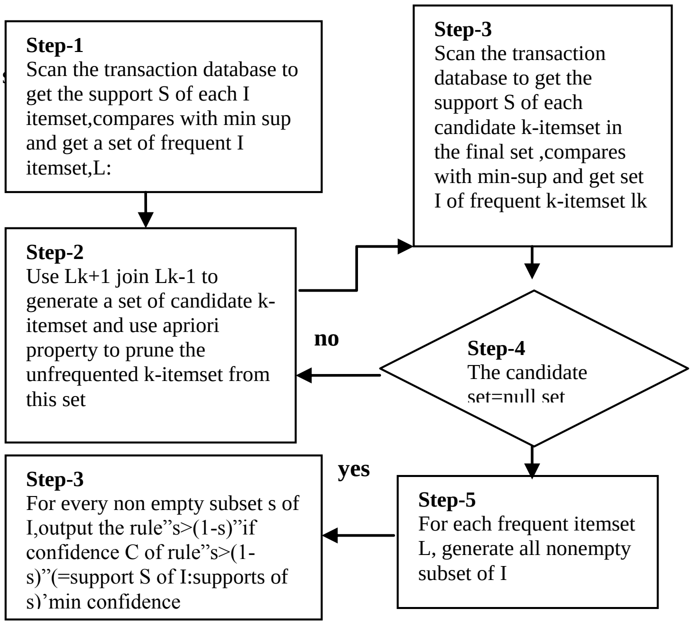

Figure 1. Apriori algorithm

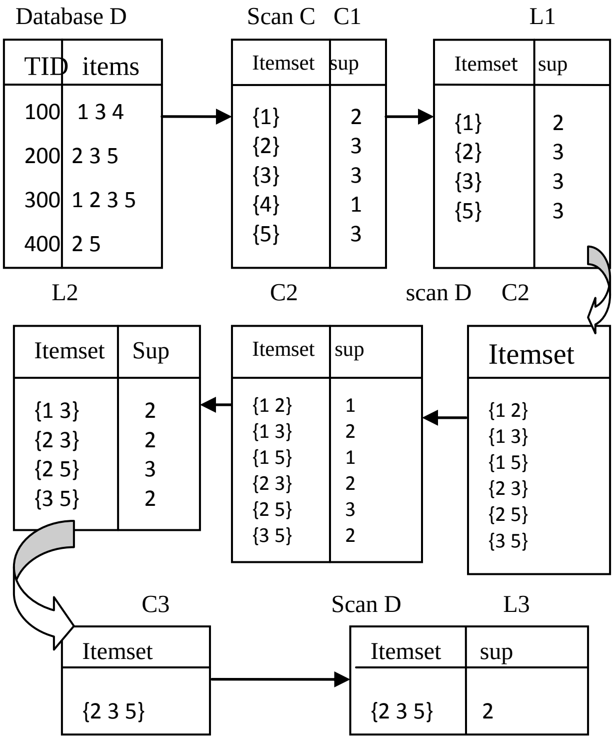

Figure 2. Apriori Example

In the mainly straight forward version of the algorithm, each item set present in any of the tuples will be measured in one pass, terminating the algorithm in one pass. In the worst case, this approach will need setting up 2^m counters equivalent to all subsets of the set of items D, where m is number of items in D. This is, of course, not only infeasible (m can easily be further than 1000 in a supermarket setting) but also unnecessary. Certainly, most likely there will extremely few large item sets containing more than l items, where 1 is small. Hence, a lot of those 2^m combinations will turn out to be small in any case.

3.2 Bottlenecks of the Apriori Algorithm

In Apriori algorithm there are two bottlenecks.

- One is the complex applicant generation process that utilizes most of the time, space and memory.
- Another bottleneck is the multiple scan of the database. Based on Apriori algorithm.

Above instance shows the working of Apriori algorithm. In every pass of the algorithm item sets of diverse size are generated. To analyze support count for each item set multiple passes to the dataset is necessary so the time taken by process to evaluate support count is more and is keep on increasing as the size of the dataset enhanced.

4. TOPICAL ALGORITHM FOR FREQUENT ITEM SET

Topical algorithm [17] are Integrated approach of Parallel Computing and ARM for mining Association Rules in Generalized data set that is basically different from all the earlier algorithms in that it utilizes database in transposed form and database transposition is done utilizing Parallel transposition algorithm (Mesh Transpose) so to generate every important association rules number of passes required is decreased. We will evaluate proposed algorithm with Apriori algorithm for frequent item sets generation. The CPU and I/O overhead can be decreased in our proposed algorithm and it is much quicker than other Association Rule Mining algorithms.

Table1: Comparison of Apriori with Topical Algorithm

Algorithm	**Data Preprocessing**	**Scan**	**Data Set**
Apriori	No Facility	Repeated Scan	Boolean
Topical Algorithm	Parallel Preprocessing	One Time Scan	Boolean

Transaction database

	A1	**A2**	**A3**	**A4**	**A5**
T1	10	0	0	0	10
T2	0	20	0	20	0
T3	0	0	0	30	50
T4	0	30	40	0	0
T5	0	0	0	0	50

Table -1

Parallel Transposition

Logical Rowwise

A1	10	0	0	0	0
A2	0	20	0	30	0
A3	0	0	0	40	0
A4	0	20	30	0	0
A5	10	0	50	0	20

1
2
1
2

Table-2

Logical Dot Product

Logical Rowwise Sum

A1*A2	0	0	0	0	0
A1*A3	0	0	0	0	0
A1*A4	0	0	0	0	0
A1*A5	1	0	0	0	0
A2*A3	0	0	0	1	0
A2*A4	0	1	0	0	0
A2*A5	0	0	0	0	0
A3*A4	0	0	0	0	0
A3*A5	0	0	0	0	0
A4*A5	0	0	1	0	0

0
0
0
1
1
1
0
0
0
1

Table 3

Figure 3. Illustration of Topical Algorithm

5. PROBLEM IDENTIFICATION

We can summarize the working of topical algorithm as follows.

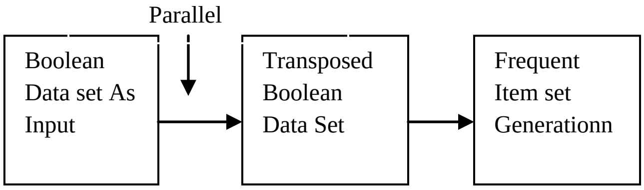

Figure 4. Topical Algorithm Working

Topical algorithms utilizes Boolean data set as input but the transaction data set are not in Boolean data type hence there is separate application necessary that will convert Generalized data set into Boolean data set which will reduce the overall efficiency of topical algorithm. As we are seen in Table 1 topical algorithm is efficient than classical Apriori algorithm. The most important benefit of topical proposed algorithm is as follows:-

- Applicant generation becomes easy and fast.
- Association rules are produced much quicker, since retrieving a support of an item set is quicker.
- The original document is not influenced by the pruning process where its role ends as soon as data is stored in 2-d array.
- The retrieval of support of an item set is faster.

Topical algorithm utilizes Parallel Mesh Transpose for transposition of 2d array. Since speeding up computations appears to be the main cause behind our interest in building parallel algorithm, the most significant measure in calculating a parallel algorithm is therefore its running time. This is defined as the time taken by the algorithm to resolve a problem on a parallel computer, that is, the time elapsed from the moment the algorithm starts to the moment it terminates.

In calculating a parallel algorithm for a given problem, it is quite natural to do it in terms of the best accessible sequential algorithm for that problem. Thus a superior indication of the quality of a parallel algorithm is the speed up it produces. This is defined as

$$\text{Speed Up} = \frac{\text{Worst} - \text{case running time of fastest known sequential algorithm for problem}}{\text{Worst} - \text{case running time of parallel algorithm}}$$

We that topical algorithm utilizes Mesh Parallel transpose for 2D array transposition. MESH TRANSPOSE calculate the transpose of an n x n matrix in O(n) time. We also noted that this running time is the quickest that can be obtained on a mesh with one data element per processor. However, since the transpose can be calculate sequentially in $O(n^2)$ time, the speed up accomplished by procedure MESH TRANSPOSE is only linear. This speed up may be considered rather small since the process utilizes a quadratic number of processors i.e. if same number of processors arranged in a unlike geometry can transpose a matrix in logarithmic time.

6. CONCLUSION

ARM algorithms are significant to find out frequent item sets and patterns from big databases. In this paper, we have studied classical and topical algorithms for generation of frequent item sets all are similar to Apriori algorithm. Topical algorithm can progress the efficiency of Apriori algorithm and it is observed to be extremely fast. Still there are few problems which we have discussed in Problem identification part i.e. Topical algorithms utilizes Boolean data set as input but transaction data set are not in Generalized data type and hence there is separate application needed that will convert Generalized data set into Boolean data set which will reduce the overall efficiency of topical algorithm and we can also enhance the efficiency of parallel transposition algorithm.

7. REFERENCES

[1] C.-Y. Wang, T.-P. Hong and S.–S. Tseng. “Maintenance of discovered sequential patterns for record deletion”. Intell. Data Anal. pp. 399-410, February 2002.

[2] R. Agrawal, T. Imielinksi and A. Swami, “Database Mining: a performance perspective”, IEE Transactions on knowledge and Data Engineering, 1993

[3] Agrawal, R., Imielinski, T., and Swami, A. N. “Mining Association Rules between Sets of Items in Large Databases”. Proceedings of the ACM SIGMOD, International Conference on Management of Data, pp.207- 216, 1993.

[4] Agrawal. R. and Srikant. R., “Fast Algorithms for Mining Association Rules”, Proceedings of 20th International Conference of Very Large Data Bases. pp.487-499, 1994.

[5] Jong Park, S., Ming-Syan, Chen, and Yu, P. S. “Using a Hash-Based Method with transaction Trimming for Mining Association Rules”. IEEE Transactions on Knowledge and Data Engineering, 9(5), pp.813-825, 1997.

[6] M. H. Margahny and A. A. Mitwaly, “Fast Algorithm for Mining Association Rules” in the conference proceedings of AIML, CICC, pp(36-40) Cairo, Egypt, 19-21 December 2005.

[7] Y. Fu., “Discovery of multiple-level rules from large databases”, 1996.

[8] F. Bodon, “A Fast Apriori Implementation”, in the Proc.1st IEEE ICDM Workshop on Frequentc Itemset Mining Implementations (FIMI2003, Melbourne,FL).CEUR Workshop Proceedings 90, A acheme, Germany 2003.

[9] Akhilesh Tiwari, Rajendra K. Gupta, and Dev Prakash Agrawal, “Cluster Based Partition Approach for Mining Frequent Itemsets” in the International Journal of Computer Science and Network Security(IJCSNS), VOL.9 No.6,pp(191-199) June 2009.

[10] JiaWei Han Micheline Kamber.”Data Mining:Concepts and Techniques”[M].Translated by Ming FAN, XaoFeng MENG etc. mechanical industrial publisher,BeiJing,2001,150-158.

[11] M. J. Zaki. “Scalable algorithms for association mining”. IEEE Transactions on Knowledge and Data Engineering, 12: 372 –390, 2000.

[12] Jochen Hipp, Ulrich G¨untzer, and Gholamreza Nakhaeizadeh. “Algorithms for Association Rule Mining – A General Survey and Comparison”.ACM SIGKDD, July 2000, Vol-2, Issue 1, page 58-64.

[13] Sotiris Kotsiantis, and Dimitris Kanellopoulos. “Association Rules Mining: A Recent Overview”. GESTS International Transactions on Computer Science and Engineering, Vol.32 (1), 2006, pp. 71-82.

[14] S. Brin, R. Motwani, J. D. Ullman, and S. Tsur, “Dynamic itemset counting and implication rules for market basket data”, SIGMOD Record 26(2), pp. 255–276, 1997.Kim Man Lui, Keith C.C. Chan, and John Teofil Nosek “The Effect of Pairs in Program Design Tasks” IEEE transactions on software engineering, VOL. 34, NO. 2, march/april 2008.

[15] Eui-Hong Han, George Karypis, and Kumar, V. Scalable “Parallel Data Mining for Association Rules”. IEEE Transaction on Knowledge and Data Engineering, 12(3), pp.728-737, 2000.

[16] Sanjeev Kumar Sharma and Ugrasen Suman “A Performance Based Transposition Algorithm for Frequent Itemsets Generation” International Journal of Data Engineering (IJDE), Volume (2) : Issue (2) : 2011

[17] Sujni Paul “An Optimized Distributed Association Rule Mining Algorithm In Parallel and Distributed Data

[18] Sujni Paul “An Optimized Distributed Association Rule Mining Algorithm In Parallel and Distributed Data Mining With Xml Data For Improved Response Time”.International Journal Of Computer Science And Information Technology, Volume 2, Number 2, April 2010

[19] Manoj Bahel and Chhaya Dule “Analysis of frequent item set generation process in Apriori & RCS (Reduced Candidate Set) Algorithm” National Conference on Information and Communication Technology, Banglore April 2010

[20] Sedukhin, S.G., Zekri, A.S. and Myiazaki, T.”Orbital Algorithms and Unified Array Processor for Computing 2D Separable Transforms” Parallel Processing Workshops