An Efficient Association Rule Mining by Optimal Multiple-Core Algorithm

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 aims to extract frequent patterns, interesting correlations, associations or casual structures among the sets of objects in the transaction files or from the other data repositories. It plays a vital role in spawning frequent item sets from large transaction databases. The discovery of interesting association relationship among business transaction records in many commercial decision making method such as catalog decision, cross-marketing, and loss-leader analysis. It is also used to excerpt hidden information from large datasets. The Association Rule Mining algorithms such as Apriori, FP-Growth wants repetitive scans over the entire file. All the input/output overheads that are being generated during the frequent perusing process, entire file decreases the performance of CPU, memory and I/O overheads. In this paper we have proposed An Cohesive tactic of Parallel Processing and ARM for mining Association Rules on Generalized data set that is basically altered from all the previous algorithms in that it uses database in transposed form and database rearrangement is done using Parallel rearrangement algorithm (Shuffle Transpose) so to generate all important association rules number of passes essential is abridged. Equaled various classical Association Rule Mining algorithms and topical procedures.

Keywords

Data Mining, Association Rule Mining (ARM), Association rule, Apriori algorithm, Frequent patterns.

1. INTRODUCTION

The fast expansion of computer technology, specially increased capacities and reduced costs of storage media, has led dealings to store vast amounts of external and internal information in big databases at low cost. Mining useful data and useful knowledge from these large databases has thus evolved into an important research area [3, 2, and 1].

Association rule mining (ARM) [18] has converted to one of the center data mining tasks and has attracted tremendous concern amongst data mining researchers. ARM is an undirected or unsupervised data mining techniques which work on variables length data, and yields clear and logical outcomes. Association Rule Mining (ARM) algorithms [17] are defined into two groups; namely, algorithms correspondingly with candidate generation and algorithms without candidate generations. In the initial group, those algorithms which are parallel to Apriori algorithm for candidate generation are considered. ECLAT might also be measured in the first group [8]. In the second category, the FP-Growth algorithms are the best–known algorithm.

The main disadvantage of previous algorithms is the recurrent scan over big database. This may be a cause of decrement in CPU presentation, recollection and increase in I/O overheads. The performances and efficiency of ARM algorithm mostly depend on three factors; viz. candidate sets engendered, data structure utilized and details of implementations [8]. ARM is an aimless or unconfirmed data mining technique which works on variable length data, and produces clear and comprehensible outcomes. Association Rule Mining (ARM) algorithms [17] are defined into two categories; namely, algorithms correspondingly with applicant generation and algorithms without candidate generation. In the initial category, those algorithms which are alike to Apriori algorithm for applicant generation are considered. Eclat might also be considered in the first group [8]. In the second group, the FP-Growth algorithm is the best–known algorithm. Following table defines the comparison among these three algorithms [9].

Table 1 – Comparison of Apriori, Eclat and FP-Growth algorithms

Algorithm	Scan	Data Structures
Apriori	M+1	Hash Table & Tree
Eclat	M+1	Hash Table & Tree
FP-Growth	2	Prefix Tree

The main drawback of above discussed algorithms is the repeated scans of large database. This may be a cause of decrement in CPU presentation, remembrance and increment in I/O expenses. The performance and efficiency of ARM algorithms mainly rests on on three issues; specifically candidate sets generated, data structure used and details of implementations [8].

The residue of this paper is structured as follows: Section 2 provides a brief review of the related work. In Section 3, we clarify common Item set and Association Rule Mining through Apriori Algorithm. In Section 4, we have explained the problematic topical algorithm and how efficacy of similar procedure can be measured and how accelerate is decided. In section 5 we have concluded our study.

2. RELATED WORK

One of the utmost well-known and common data mining techniques is the Association rules or frequent item sets mining algorithm. The algorithm was initially planned by Agrawal et al. [4] [5] for market basket analysis. Because of its significant applicability, many reviewed algorithms have been presented since then, and Association rule mining is still a widely studied zone.

Agrawal et al. presented an AIS algorithm in [4] which generates candidate item sets on-the-fly during each permit of the database scan. Huge item sets from previous pass are checked if they are present in the current transaction. Thus new article sets are shaped by extending existing item sets. This algorithm turns out to be ineffective because it produces too many applicant item sets. It requires more space and at the same time this algorithm requires too many authorizations over the entire database and also it produces rules with one consequent item.

Agrawal [5] Developed various forms of Apriori algorithm as Apriori, Apriori Tid, and Apriori Hybrid. Apriori and Apriori Tid generate item sets by means of the big item circles found in the previous pass, without considering the transactions. Apriori Tid improves Apriori by means of the database at the first pass. Counting in subsequent passes is done using encodings created in the first pass, which is minor than the file. This leads to a dramatic performance improvement of three times quicker than AIS.

Scalability is extra significant area of data mining because of its huge size. Hence, algorithms must be accomplished to “scale up” to grip vast amount of data.

Eui-Hong et al. [16] tried to make data distribution and applicant distribution accessible by Intelligent Data Distribution (IDD) algorithm and Hybrid Distribution (HD) algorithm respectively. IDD addresses the subjects of communication overhead and redundant computation by using aggregate memory to partition contenders and change data efficiently. HD advances over IDD by dynamically partitioning the candidate set to maintain good load balance. Different works are described in the literature to amend the Apriori logic so as to improve the efficiency of generating rules. These means even nevertheless focused on decreasing time and space, in real time still wants improvement.

3. FREQUENT ITEM SET AND ASSOCIATION RULE

The aim of Association rule mining is discovering associations and significant guidelines in large datasets. A dataset is considered as a sequence of entries containing of quality morals also recognized as items. A set of such item sets is called an item set. Repeated item sets are sets of pages which are visited often together in a single server period.

Let I ={ I1, I2, … , Im }be a set of objects. Let D, the task-relevant data, be a set of file dealings where each deal T is a set of objects such that T $\subseteq$ I. Each operation is connected with an identifier, named TID. Let A be a set of items. A transaction T is supposed to comprise A if and merely if A $\subseteq$ T. An association law is an effect of the form A$\Rightarrow$B, wherever A$\subset$I, B$\subset$I, and A$\cap$B=$\emptyset$. The rule A$\Rightarrow$B embraces in the transaction set D with sustenance s, where s is the percentage of dealings in D that comprise A$\cup$B (i.e., the union of set A and B, or say, both A and B). This is acquired to be the probability, P(A$\cup$B). The rule A $\Rightarrow$ B has confidence c in the transactions set D, where c is the percentages of transactions in D comprising A that also comprise B. This is acquired 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 referred to as an item set. An item set that comprises k items is a k-item set. The set {bread, butter} is a 2-itemsets. The occurrences frequency of an item set is the number of dealings that comprise the item set, it is also known, as the frequency, or support count. If the comparative provision of an item set I satisfy a pre definite minimum support threshold then I is a frequent item set. The set of recurrent k-item sets is usually denoted by Lk.

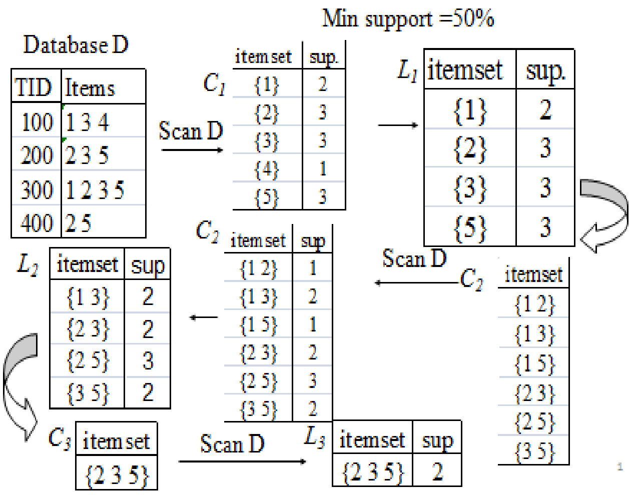
Confidence (A→B) = P(B|A) = support(AUB) / support(A) = support_count(AUB) / support_count (A)

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

By an association law, we mean an consequence of the form X$\Rightarrow$Ij, where X is a set of some items in τ, and Ij is a sole item in τ that is not existing in X. The rule X $\Rightarrow$Ij is satisfied in the set of transactions T with the confidence factor 0≤ c ≤ 1 if at least c% of dealings in T that satisfy X also satisfy Ij. We will use the notation X $\Rightarrow$Ij | c to stipulate that the rule X $\Rightarrow$Ij has a sureness factor of c[3].

3.1 Apriori Algorithm

The Apriori algorithm is one of the utmost widespread algorithms for mining frequent patterns and association rules [4]. It introduces a method to produce candidate item sets Ck in the pass k of a transaction database using only frequent item set Lk−1 in the previous pass. The idea time-outs on the fact that any subset of a recurrent item set must be frequent.



TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

itemset
{2 3 5}

itemset	sup
{2 3 5}	2

Figure 1. Apriori Example

In the utmost forthright version of the algorithm, every item set present in any of the tuples will be measured in one license, dismissing the algorithm in one pass. In the worst case, this approach will need setting up 2^m counters corresponding 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 simply be more than 1000 in a superstore setting) but also unnecessary. Indeed, most likely there will very infrequent large item sets comprising more than l items, where 1 is small. Hence, a batch of those 2^m combinations will turn out to be small anyway.

3.2 Bottlenecks of the Apriori Algorithm

In Apriori algorithm there are two bottlenecks.

- One is the complex candidate generation process that uses most of the time, spaces and memory.
- Another bottleneck is the multiples scan of the database. Based on Apriori algorithm.

Above example shows the working of Apriori algorithm. In each pass of the algorithm item sets of different size are generated. To compute support_count for each itemset multiple passes to the dataset is required so the time taken by process to calculate support_count is more and is keep on increasing as the size of the dataset increases.

3.3 Topical Algorithm For Frequent Item Set

Topical algorithm [17] are Integrated approach of Parallel Computing and ARM for mining Association Rule in Generalized data set that is fundamentally diverse from all the previous algorithms in that it uses database in transposed form and database transposition is done utilizing Parallel transposition algorithm (Mesh Transpose) so to generate all well. Hence, Ck can be generated by assembly two itemsets in Lk−1 and lopping those that contain any subset.

Table 2: 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

4. PROBLEM IDENTIFICATION

We can summarize the working of topical algorithm as follows:

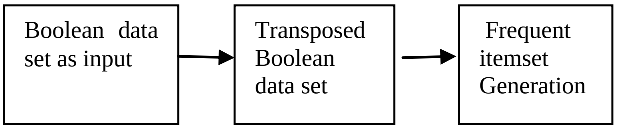

Boolean data set as input
Transposed Boolean data set
Frequent itemset Generation

Figure 2. Topical Algorithm Working

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

- Candidate generation becomes easy and fast.
- Association rules are produced much faster, since retrieving a support of an item set is quicker.
- The original file isn't influenced by the pruning process where its roles end as soon as data is stores in 2-d array.
- The retrieval of support of an item set is quicker.

Topical algorithm uses Parallel Mesh Transpose for transposition of 2d array. Since speeding up computations appears to be the major reason behind our interest in building parallel algorithm, the most important measure in evaluating a parallel algorithm is therefore its running times. This is defined as the time taken by the algorithm to solve a problem on a parallel computer, that is, the times beyond 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 available sequential algorithms for that trouble. Thus a good indication of the quality of a parallel algorithm is the speedup it produced. This is defined as

Speed Up=

$$\frac{\textbf{Worst-case running time of fastest known sequential algorithm for problem}}{\textbf{Worst-case running time of parallel algorithm}}$$

We that topical algorithm uses Mesh Parallel transpose for 2D array transposition. MESH TRANSPOSE calculated 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 computed sequentially in O(n^2) time, the speedup achieved by procedure MESH TRANSPOSE is only linear. This speedup may be considered rather small since the procedure utilize a quadratic number of processors i.e. if same number of processors arranged in a different geometry can transposes a matrix in logarithmic time.

5. METHODOLOGY AND EXPERIMENTAL RESULT

As we seen in our problem identification section topical algorithm is efficient then classical apriori algorithm but MESH Transpose distributed algorithm is not optimal which is used by topical algorithm. Mesh Transpose drawback can be overcome by Shuffle Transposition.

5.1 Proposed Algorithm

```
Procedure EPTA()
1.  Shuffle Transpose(DataSet)//Transpose the transactional database
2.  Read the database to count the support of C1 to determine L1 using sum of rows.
3.  $L_1$= Frequent 1- itemsets and k:= 2
4.  While (k-1 ≠ NULL set) do
Begin
        $C_k$: = Call Gen_candidate_itemsets ($L_k$-1)
        Call Prune ($C_k$)
        for all itemsetsi□ I do
        Calculate the support values using dot-multiplication of array;
        $L_k$ := All candidates in Ck with a minimum support;
        k:=k+1End
5. End of step-4
End Procedure
```

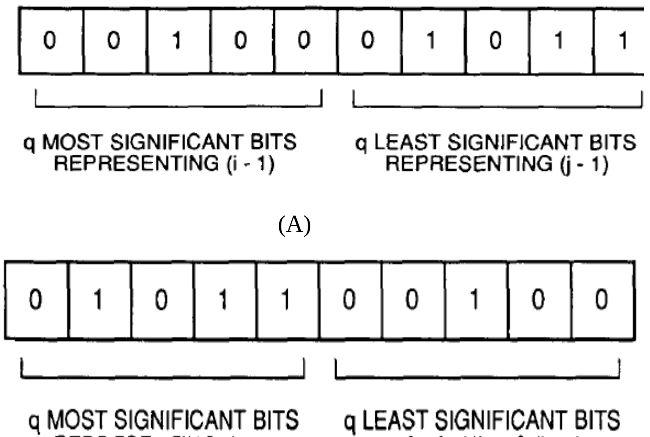
```
Procedure SHUFFLE TRANSPOSE (A)
fori= I to q do
    fork = I to 22q - 2 do in parallel
        Pk sends the element of A it currently holds to P2kmod(22q- 1)
    end for
end for
End
Procedure Gen_candidate_itemsets (Lk-1)
Ck= Φ
for all itemsets $I_1 \in L_{k-1}$ do
for all itemsets $l_2 \in L_{k-1}$ do
if $I_1[1] = I_2[1] \wedge I_1[2] = I_2[2] \wedge \ldots \wedge I_1[k-1] < I_2[k-1]$ then
    c = $I_1[1], I_1[2] \ldots I_1[k-1], I_2[k-1]$
Ck = $C_k$□ □ {c}
End Procedure
Procedure Prune(Ck)
forall c ∈Ck
forall (k-1)-subsets d of c do
ifd ∉Lk-1
thenCk= Ck– {c}
End Procedure
```

5.1.1 Shuffle Transpose

Consider a processor index k consisting of 2q bits. If k = 2q(i - 1) + (j - 1), then the q most important bits of k represent i - I while the q least significant bits represent] - 1. This is illustrated in Figure 3(A).for q = 5, i = 5, and j = 12. After q shuffle (i.e., q cyclic shifts to the left), the element originally held by P_k will be in the processor whose index is

$$s = 2^q(j - 1) + (i–1)$$

As shown in Figure 3(B). In other words a_{ij} has been moved to the position originally occupied by a_{ji}.


0 0 1 0 0 0 1 0 1 1
q MOST SIGNIFICANT BITS REPRESENTING (i - 1)
q LEAST SIGNIFICANT BITS REPRESENTING (j - 1)
0 1 0 1 1 0 0 1 0 0
q MOST SIGNIFICANT BITS REPRESENTING (j - 1)
q LEAST SIGNIFICANT BITS REPRESENTING (i - 1)

(A)

(B)

Figure 3. Derivation of number of shuffles required to transpose matrix

6. EXPERIMENTAL RESULT

The program for Apriori and our proposed algorithm were developed in Java JDK1.5 environment and for Distributed Shuffle Transpose algorithm have used MPI (Message Passing Interface).The Open MPI Project is an open resource Message Passing Interface implementation that is developed.

7. CONCLUSION

ARM algorithms are significant to discover frequent item sets and patterns from large 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 very quick. Still there are some problems which we have discussed in Problem identification section i.e. Topical algorithms uses Boolean data set as input but the transaction data set are not in Boolean data type hence there is separate application required that will convert Generalized data set into Boolean data set which will decrease the overall efficiency of topical algorithm. Shuffle parallel transposition algorithm which an Optimal parallel transposition having time complexity O (log In the utmost forthright version of the algorithms, each item set present in any of the tuples will be measured in one license, dismissing the algorithm in one pass. In the worst case, this approach will need setting up 2^m counters corresponding 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 more than 1000 in a superstore setting) but also redundant. Indeed, most likely there will very infrequent large item sets comprising more than l items, where 1 is small. Hence, a lots of those 2^m combination will turn out to be small anyway.

8. 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] M.S. Chen, J.Han and P.S. Yu. “Data Mining: An overview from a database perspective”, IEE Transactions on Knowledge and Data Engineering 1996.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

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

[10] 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.

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

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

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

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

[15] 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 TeofilNosek “The Effect of Pairs in Program Design Tasks” IEEE transactions on software engineering, VOL. 34, NO. 2, march/april 2008.

[16] 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.

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

[18] Dr (Mrs).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] ManojBahel, ChhayaDule “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.; Myiazaki, T.”Orbital Algorithms and Unified Array Processor for Computing 2D Separable Transforms” Parallel Processing Workshops (ICPPW), 2010 39th International Conference Page(s): 127 – 134.