Volume 4, No. 8, May-June 2013
ISSN No. 0976-5697
International Journal of Advanced Research in Computer Science
REVIEW ARTICAL
Available Online at www.ijarcs.info

Algorithms for Association Rule Mining:A General Survey on Benefits And Drawbacks of Algorithms

Venu Mishra 1st
Computer Science Department
Jagannath University
Jaipur, India
venu0110mishra@gmail.com

Tarun Kumar Mishra 2nd
Computer Science Department
Poornima College of Engineering
Jaipur, India
mishratarun2005@gmail.com

Arun Mishra 3rd
Computer Science Department
Maharshi Arvind Institute of Engineering and Technology
Jaipur, India
mishra.arun.mishra29@gmail.com

Abstract: This In this paper, we provide a recent review of the association rule mining algorithms. Of course this article is not enough to provide detail of all association rule mining algorithms. But it enlighten the major issues of the mining algorithms. Association rule mining is the process of finding the relationship between items of the database. In this paper we will cover the major theoretical issues, guiding the researcher in the directions that have yet to explore.

Keywords: Association rule, relational database, WARM,HITS,SEE,DHP

I. INTRODAUCTION TO MINING

As the technology increase new dating techniques like cloud computing and service oriented architecture need for integrating scattered data and finding interesting information out of that is a new growing challenge. This information cannot be extracted by the traditional methods such as queries or statistical analysis. This information can be used in classification of the data and in prediction of future events etc. So far many techniques have been implemented that are being going to use in data mining. Association rule mining is one of the techniques. But there are some challenges associated with ARM improving processing speed and reducing communication. Association rule mining is to find the interesting patterns from the large database for further analysis. [1]

II. PROBLEM DEFINITION

We can define association mining problem as: Let DB_T is a database of transactions, each transaction consists of I, where I is $\{i_1,i_2,i_3 \ldots i_n\}$ items. Association rules are in the form of $A \rightarrow B$, where $A \subset I, B \subset I \text{ and } A \cap B = \phi$ .Each association rule have support and confidence that specify the significance of the association rule. Support denotes the occurrence of item in the database DB_T. Confidence is the proportion of the data items containing B in all the items containing A also in DB_T.[1]

$$Sup(A) = Count(A) / Count(DB_T)$$

$$Sup(A \rightarrow B) = Sup(A \cup B)$$

$$Conf(A \rightarrow B) = Sup(A \cup B) / Sup(A)$$

If the support and confidence of the rule is greater or equal to the threshold value minsup and min conf then the rule is consider as a valid rule otherwise it is discarded. The objective of the ARM is to find the set of the valid association rules.

The most common approach to finding association rules is to break up the problem into two parts :

a. Find all frequent itemsets: By definition, each of these itemsets will occur at least as frequently as a pre-determined minimum support count .
b. Generate strong association rules from the frequent itemsets: By definition, these rules must satisfy minimum support and minimum confidence .

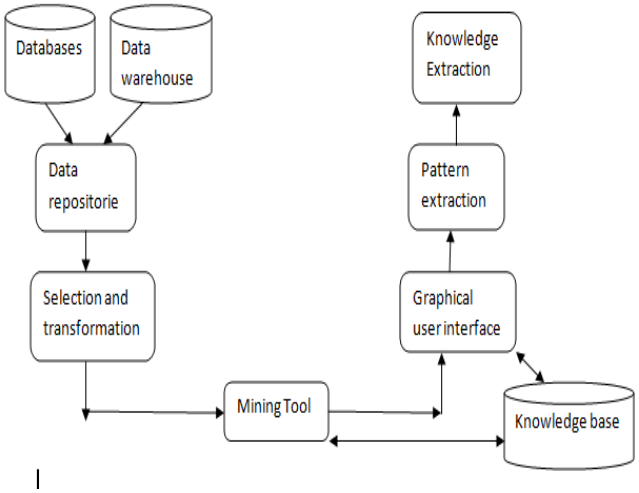

Figure: process of Mining

Generally, an association rules mining algorithm contains the following steps

a. First one, k itemsets is generated by the large (k 1)itemsets generated in the previous iteration.
b. Second, Supports for the candidate k itemsts are generated by a database scan.
c. At last Itemsets that do not have the minimum support are discarded and the remaining itemsets are called large k itemsets and considered for next iteration of finding frequent itemset.

This process is repeated until no more large itemsets are found.

III. CLASSICAL ALGORITHMS FOR MINING

There are many algorithms are proposed to perform ARM on transactional databases, Comparison analysis of some of them is described below:

A. AIS Algorithm[1]:

Steps followed in AIS algorithm

a) Candidate item sets are generated and counted on-the-fly as the database is scanned.
b) For each transaction, it is determined which of the large item sets of the previous pass are contained in this transaction.
c) New candidate item sets are generated by extending these large item sets with other items in this transaction.

a. AIS mining process:

In AIS, the frequent item sets were generated by scanning the databases several times. The support count of each individual item was accumulated during the first pass over the database. Based on the minimal support count those items whose support count less than its minimum value gets eliminated from the list of item. Candidate 2-itemsets are generated by extending frequent 1-itemsets with other items in the transaction. During the second pass over the database, the support count of those candidate 2-itemsets are accumulated and checked against the support threshold. Similarly those candidate (*k+1*)-item sets are generated by extending frequent *k*-item sets with items in the same transaction. All that candidate item sets generation and frequent item sets generation process iterate until any one of them becomes empty. [1]

b. Drawbacks:

The main drawbacks of the AIS algorithm are

a) Too many candidate item sets that finally turned out to be small are generated, which requires more space and wastes much effort that turned out to be useless.
b) AIS algorithm results in unnecessarily generate and count too many candidate item sets that turn out to be small.
c) This algorithm requires too many passes over the whole database.

B. SETM (Set Oriented Mining)[1]:

The SETM algorithm [5]is better than AIS algorithm. This algorithm uses only standard database operations to find frequent sets. For this reason, it uses its own data representation to store every itemset supported by a transaction along with the transaction's ID (TID). In SETM database is modified repeatedly to perform candidate generation, support counting, and remove infrequent sets. Fewer candidate generation is the advantage of SETM over AIS.

a. Drawbacks:

However, the problem with the SETM algorithm is that candidates are replicated for every transaction in which they occur, which results in huge sizes of intermediate results. Moreover, the itemsets have to be stored explicitly, i.e., by listing their items in ascending order. Using candidate IDs would save space, but then the join could not be carried out as an SQL operation. What is even worse is that these huge relations have to be sorted twice to generate the next larger frequent sets.

C. Apriori Algorithm:

Apriori algorithm was first proposed by Agarwl. Apriori is more efficient the candidate generation process is more efficient[2]. To count the support of the item sets it uses the BFS technique. And uses a candidate generation function which exploits the downward closure property of support.

Apriori algorithm state that itemset x containing subset itemset y is never frequent if y is not frequent based on this principle, Apriori generate new itemset of length K+1 using k frequent itemset.and eliminate rest elements those have infrequent itemset. So Apriori generates new itemsets by using frequent itemsets of previous itemset without considering transaction. But it consider transaction for counting of support of the new candidates.[1,2]

Steps involved in Apriori algorithm

a) Candidate item sets are generated using only the large item sets of the previous pass without considering the transactions in the database.
b) The large item set of the previous pass is joined with itself to generate all item sets whose size is higher by 1.
c) Each generated item set that has a subset which is not large is deleted. The remaining item sets are the candidate ones.

a. Drawbacks:

The main drawbacks of Apriori algorithm are

a) It takes more time, space and memory for candidate generation process.
b) To generate the candidate set it requires multiple scan over the database.

D. FP-Tree Algorithm:

FP-Tree frequent pattern mining is used in the development of association rule mining. FP-Tree algorithm overcomes the problem found in Apriori algorithm. The frequent item set generation process requires only two passes over the database there is no need for candidate generation

process. By avoiding the candidate generation process and less passes over the database, FP-Tree founds to be faster than the Apriori algorithm. An FP-Tree is a prefix tree for transactions. Every node in the tree represents one item and each path represents the set of transactions that involve with the particular item. All nodes referring to the same item are linked together in a list, so that all the transactions that containing the same item can be easily found and counted.[1,2]

a. Algorithm Process:

FP-tree algorithm involves the generation of frequent patterns using the frequent patterns generation process which includes two sub processes:

(a). Constructing the FP-Tree, and

(b). Generation of frequent patterns from theFP-Tree.

The efficiency of FP-Tree algorithm account for three reasons[1]

a) FP-Tree is a compressed representation of the original database because only those frequent items are used to construct the tree, other irrelevant information are pruned. Also by ordering the items according to their supports the overlapping parts appear only once with different support count.

b) This algorithm only scans the database twice. The frequent patterns are generated by the FPgrowth procedure, constructing the conditional FPTree which contain patterns with specified suffix patterns, frequent patterns can be easily. Also the computation cost decreased dramatically.

c) FP-Tree uses a divide and conquers method that considerably reduced the size of the subsequent conditional FP-Tree, longer frequent patterns are generated by adding a suffix to the shorter frequent patterns.

b. Drawbacks:

The disadvantages of FP-Tree Algorithm are

(a). It requires several scan over the database for the construction of FP-Tree.

(b). Whole mining process should be repeated whenever the support value is changed as well when a new dataset is inserted into the database.

IV. IMPROVEMNET IN CLASSICAL ALGORITHMS

A. AprioriTid & AprioriHybrid:

This algorithm is use the concept of the Apriori but it is good over Apriori in term of database scan. It make TIDs at each pass. It just scan database in first pass. Aafter that it use Tid's for counting.but the major downside of this algorithm is the generation of the Tid table at each pass.

AprioriTid can be considered an optimised version of SETM that does not rely on standard database operations and uses *apriori-gen* for faster candidate generation. Therefore, comparing *Apriori* and *AprioriTid* is more interesting because they both generate the same number of candidates and differ mainly in their underlying data representation.

Another algorithm, called Apriori Hybrid, is introduced in. This use the combination of Apriori and aprioriTid. Idea behind this algorithm is to run the Apriori algorithm initially, when transactions are large and then switch to the AprioriTid algorithm when the generated database, i.e. large k itemset in the transaction with identifier TID, would fit in the memory.

a. Drawbacks:

a) The size of database is limited to the main memory size.

b) Second problem is the pruning of the database in the later stages of the algorithm. i.e. the removing the part that will not be used further for mining process.

B. DHCP Algorithm:

The DHP (Direct Hashing and Pruning) algorithm is an effective hash based algorithm for the candidate set generation. It reduced the size of candidate set by filtering any k itemset out of the hash table if the hash entry does not have minimum support. The hash table structure contains the information regarding the support of each itemset. The DHP algorithm consists of three steps. The first step is to get a set of large 1-itemsets and constructs a hash table for 2itemsets. The second step generates the set of candidate itemsets Ck. The third step is the same as the second step except it does not use the hash table in determining whether to include a particular itemset into the candidate itemsets. Furthermore, it should be used for later iterations when the number of hash buckets with a support count greater than or equal to the minimum transaction support required is less than a predefined threshold.[5,6]

C. Partitioning Algorithms:

Algorithms so far discussed are more or less variations of the same scheme. But partitioning algorithm take different approach. Partitioning algorithm provide the method to overcome from the two major shortcomings of the previous algorithms. First one is that in the previous algorithm the passes over the database are not known beforehand, regardless of which representation is used. ApriroiTId tries to recover this problem but in that case the size of database is limited to the main memory size. Second problem is the pruning of the database in the later stages of the algorithm. i.e. the removing the part that will not be used further for mining process.

This problem can't be solved by the above mentioned techniques .Partitioning algorithm solve first problem by partitioning the database into equally sized horizontal partitions. An algorithm that is used for mining run on the subset of the transactions independently, and produce local frequent itemset for each partition.The size of partition is depending on the size of memory so that partition can reside in the memory during run. Hence only one read is required for this step. And rest all passes get access of the buffered data.

Now come to the second problem (failure to reduce the database size in later passes), *Partition* uses a "TID-list" data representation to determine the frequent itemsets for each partition and to count the global supports during the counting phase. TID list conatains the information about all the itemsets. It stores the information of the all transactions to those the itemsetset belongs.so TID-lists can be calculated by intersacting the the TID-liste of the (k-1) itemsets. TId-lists

will change in every pass and may not be swapped if there is no enough space in memory. [5]

Partitioning may improve the performance of finding large itemsets in several ways:

(a). Partitioning take the advantage of the the large itemset properties that large itemset must in at least one partitions.so it improve the database scan process compare to the other one in which we look whole database for finding frequent itemsets.

(b). Partitioning algorithm improves the memory limitation problem of the APrioriTID algorithm.Each partition created in such a manner so that it can fit to the memory.and also it imporves the total itemset counted per itemset as compare to the whole database.

(c). One another benefit of this algorithm is that it can be applied to the other distributed algorithms. And that case each partion treated as a separate unit for process.

(d). This algorithm also help in the incremental generation of association rules. In this it will help by treating the current state of the database as one partition and the new state as the second partition.

D. Sampling:

Toivonen presented An association rule mining algorithm using sampling. the approach can be divided in to two phases. During phase one a sample of the database is extracted and mining perform on that sample to extract frequent rules .then these rules are validated by the whole database. But auther suggested that support should be take lower at first phase because itemsets those not frequent in sample may be frequent in the whole database.

Chuang et al. explore progressive sampling algorithm, that is sampling error estimation(SEE), which identified that appropriate sample size for mining association rules. First one is that SEE is highly efficient because an appropriate sample size can be determined without the need of executing association rules. the identified sample size of SEE is very accurate, meaning that association rules can be highly efficiently executed on a sample of this size to obtain a sufficiently accurate result.[5]

E. Database Partitioning:

The algorithm reduces the database activity by computing all frequent sets in two passes over the database. Into sections small enough to be handled in main memory. That is, a part is read once from the disk, and level-wise generation and evolution of candidates for that part are performed in main memory without further database activity. The first database pass consists of identifying the collection of all locally frequent sets in each database part. For the second pass, the union of the collections of locally frequent sets is used as the candidate set. The union of the collections of locally frequent sets is used as the candidate set. The first pass is guaranteed to locate a superset of the collection of frequent itemsets; the second pass is needed to merely compute the frequencies of the sets to extract the global frequent sets.

The main achievement of partition is the reduction of the database activity. It was shown that this reduction was not obtained at the expense of more CPU utilization, which is another achievement.[5,6]

F. Dynamic Itemset Counting:

Another algorithm called DIC (dynamic itemset counting) was proposed by brin et al. in 1997. It tries to reduce the database activity by counting candidate itemsets earlier than Apriori does. In Apriori, candidate (k+1)-itemsets are not counted until the $(k+1)^{st}$ pass on the database (although an optimization called pass bundling permits counting candidates of different sizes in the same pass if memory is available). In DIC, on the other hand, candidate (k+1)-itemsets are counted as soon as the algorithm discover that all its subsets of size k have exceeded the support threshold and will be frequent. This is done by stopping at various points in the database to examine the possibility of including other itemsets in the counting procedure. It has been found that such techniques, with reasonable setting of the number of transactions passed before stopping for recalculation, can reduce the number of database passes dramatically while maintaining the number of candidate sets that need to be counted relatively low compared to other proposed techniques.[5,7]

G. WARM Algorithms:

Algorithms that we have studied yet all treat the all item of the dataset as same so now we introduce some weighted association rule mining algorithms.that does not work on databases with only binary attributes.in this we use the importance of the all data of the datasets.in WARM each item has weight that refelect the importance of that item.the weight may correspond to special promotions on some products, or profitability of the products.[9]

HITS **(Hyperlink Induced Topic Search)** are the one of the example of the WARM algorithm that is used for web mining. A generalized version of the HITs is applied to the graph of the items to rank items. in the graph all links and nodes have weighted as their significance. And we we use support for items as per their significance.

This HITS algorithm is suitable only for static content.

a. They work well in environments where no dynamic updating is possible.

b. They fail to capture the rich information that lie within the patterns of user access or in the structure that can be defined by user group implicitly.[4]

Table I : Categorization of algorithms

NAME OF ALGORITHM	CATEGORY
AIS	Classical Mining Algorithm
SETM	
Apriori	
AprioriTid and AprioriHybrid	Transaction and Item Pruning algorithms
DHP	
Database partitioning	Reduced the number of passes of databases
Dynamic Itemset Counting	
Sampling Technique	

V. CONCLUSION

Information is collected almost everywhere in our everyday lives. This leads to the huge increase in the amount of data available. Physical analysis of these huge amount of information stored in modern databases is very difficult. Data mining provides tools to reveal unknown information in large databases which are stored already. A well-known data mining technique is association rule mining. Association rules are very efficient in revealing all the interesting relationships in a relatively large database with huge amount of data. This paper provides some of the existing data mining algorithms for market basket analysis. The analysis of existing algorithms suggests that the usage of association rule mining algorithms for market basket analysis will help in better classification of the huge amount of data. The apriori algorithm can be modified effectively to reduce the time complexity and enhance the accuracy. But in every algorithm there founds a common drawback of various scans over the database this drawback can be overcome by introducing a new technique of transaction pattern base which founds to be efficient for searching frequent patterns in the database.

VI. REFERENCES

[1]. R.Divya , S.Vinod kumar ," Survey On Ais, Apriori And Fp-Tree Algorithms", International Journal of Computer Science and Management Research Vol 1 Issue 2 September 2012 ISSN 2278-733X

[2]. S.Suriya, Dr.S.P.Shantharajah, R.Deepalakshmi, " A Complete Survey on Association Rule Mining with Relevance to Different Domain" International Journal Of Advanced Scientific And Technical Research Issue2, Volume 1 (February 2012) Issn: 2249-9954

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

[4]. XindongWu · Vipin Kumar," Top 10 algorithms in data mining" Published online: 4 December 2007 © Springer-Verlag London Limited 2007

[5]. Ahmed Medhat Ayad thesis on "A New Algorithm For Incremental Mining Of Constrained Association Rules" Page no.20-23

[6]. http://software.intel.com/en-s/articles/Multicoreenabling-FP-tree-Algorithm-for-Frequent-Pattern-Mining

[7]. Agrawal, Rakesh; and Srikant, Ramakrishnan;"Fast algorithms for mining association rules in large Databases"

[8]. http://chemeng.utoronto.ca/~datamining/dmc/association_rules.htm

[9]. Han, J. and Kamber, M. 2000. Data Mining Concepts and Techniques. Morgan Kanufmann.

.