# Implementation of Enhanced Association Rule Mining on Horizontal Distributed Databases

[1]Puja Anil Naval, [2]Ashwini Sudhir Patil, [3]Punam Ashok Patil, [4]Pooja Vasant Sapkale

Department of Computer Engineering
SSBT COET Bambhori, Jalgaon

***Abstract*: One of the reasons behind maintaining any database is to enable the user to find interesting patterns and trends in the data. Frequent sets play an essential role in many data mining tasks that try to find interesting patterns from databases, such as association rules, correlations and clusters. The mining of association rules is one of the most popular of all these. The proposed algorithms is a useful for discovering customer purchasing patterns by extracting associations or co-occurrences from transactional databases. The idea behind the proposed algorithms is to examine the orders for products that have been purchased together. A store could use this information to place products frequently sold together into the same area. Direct marketers could use the sales tracking results to determine what new products to offer to their prior customers. The proposed algorithms offer efficiency and enhanced privacy with respect to the current leading protocols.**

***IndexTerms*: Privacy Preserving Data Mining; Frequent Itemsets; Association Rules.**

## 1. Introduction:

The major reason that data mining has attracted a great deal of attention in information industry in recent years is due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. Frequent sets play an essential role in several data processing tasks that try and find interesting patterns from databases, like association rules, correlations, sequences, episodes, classifiers and clusters. The mining of association rules is one in all the foremost common problems of of these. The identification of sets of items, products, symptoms and characteristics, which often occur together in the given database, can be seen as one of the most basic tasks in data mining. The original motivation for searching frequent sets came from the need to analyze supermarket transaction data, that is, to examine customer behavior in terms of the purchased products. Frequent sets of products describe how often items are purchased together. Frequent sets of products describe however usually things are purchased together. Association rule mining is used to find interesting associations and/or correlation relationships among large sets of data items. In this paper, discuss the problem of computing association rules within a horizontally partitioned database. All sites have the same schema, but each site has information on different entities. The goal is to produce associate ion rules that hold globally, while limiting the information shared about each site to preserve the privacy of data in each site. Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time. Let X be an itemset, X ---> Y an association rule and T a set of transactions of a given database.

**Support**: Support is an indication of how frequently the itemset appears in the database. The support of X with respect to T is defined as the proportion of transactions t in the database which contains itemset X.

$$supp(X)= supp(X)/T$$

**Confidence:** Confidence is an indication of how often the rule has been found to be true. The confidence value of a rule, X ---> Y, with respect to a set of transactions T, is the proportion of the transactions that contains X which also contains Y. Confidence is defined as:

$$conf(X \dashrightarrow Y) = supp(X, Y)/supp(X)$$

It consists of three blocks. First step include constructing the table which stores all individually frequent items and the support of each item. Second step is pruning. Infrequent items are then removed. The unwanted transactions that contain the entire set of items which fall below the mentioned support level are removed in the start itself. After this stage complete data is cleaned as required then the actual mining process takes place where all the frequent item sets of two or more items are mined.

## 2. Objectives:

Data mining has attracted a great deal of attention in information industry in recent years is due to the wide availability of huge amounts of data and the imminent need for turning such data into useful information and knowledge. The proposed system satisfies the following objectives:-

- To make more informed decisions about product placement, pricing and profitability.
- To learn more about customer behavior.
- To find out which products perform similarly to each other.
- To determine which products should be placed near each other.

- To find out which products should be cross-sold.
- To find out if there are any successful products that have no significant related elements.

### 3. Literature Survey:

The progress in bar-code and computer technology has made it possible to collect data about sales and store as transactions which is called basket data. This stored data attracted researches to apply data mining to basket data. As a result association rules mining came into prominence which is mentioned as synonymous to market basket analysis. As stated before association rule mining is a two step process. Firstly, frequent itemsets are found using minimum support value, and this step is the main concentration of association rule mining algorithms. Later from these itemsets using minimum confidence value rules are produced. As the differing part of the algorithms are frequent itemset finding part, association rule mining, frequent itemset mining or frequent pattern mining terms are used interchangeably. Association rule mining which was first mentioned is one of the most popular data mining approaches. Not only in market business but also in variety of areas association rule mining is used efficiently. There are numerous algorithms for locating most frequent combination of items.

### Apriori Algorithm:

Apriori [5] is an algorithm for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database. The Apriori algorithm was proposed by Agrawal and Srikant in 1994. Apriori is designed to operate on databases containing transactions for example, collections of items bought by customers. Each transaction is seen as a set of items an item set. Given a threshold$C$, the Apriori algorithm identifies the item sets which are subsets of at least $C$ transactions in the database. Apriori uses a "bottom up" approach, where frequent subsets are extended one item at a time, a step known as candidate generation, and groups of candidates are tested against the data. The algorithm terminates when no further successful extensions are found. Apriori uses breadth-first search and hash tree to count candidate item sets efficiently. It generates candidate item sets of length K from item sets of length K-1. Then it prunes the candidates which have an infrequent sub pattern. According to the downward closure lemma, the candidate set contains all frequent K-length item sets. After that, it scans the transaction database to determine frequent item sets among the candidates.

### 4. Proposed System:

### 4.1. FDM Algorithm:

Proposed algorithm are based on the Fast Distributed Mining (FDM) of Cheung et al.[3], which is an unsecured distributed version of the Apriori algorithm. Its main idea is that any *s*-frequent itemset must be also locally *s*-frequent in at least one of the sites. Hence, in order to find all globally *s*-frequent itemsets, each player reveals his locally *s*-frequent itemsets and then the players check each of them to see if they are *s*-frequent also globally[3]. FDM algorithm steps are as follows:

1. Initialization
2. Candidate Sets Generation
3. Local Pruning
4. Unifying the candidate itemsets
5. Computing local supports
6. Broadcast Mining Results

For secure computation of frequent itemsets the Advanced Encryption Standard(AES) is applied on proposed algorithm which is based on FDM. AES is a symmetric key encryption standard adopted by the U.S. government. The key size used for an AES cipher specifies the number of repetitions of transformation rounds that convert the input called the plaintext into the final output called the cipher text.

### 4.2. Algorithm 1:

Algorithm 1 is important frequent pattern mining method, which generates frequent itemset without candidate generation. The Algorithm 1 adopts a divide and conquers strategy as follows: compress the database representing frequent items into a frequent-pattern tree, but retain the itemset association information, and then divide such a compressed database into a set of condition databases, each associated with one frequent item, and mine each such database. Firstly database is read and frequent items are found which are the items. are occurring in transactions less than minimum support. Secondly database is read again to build tree. After creating the root, every transaction is read in an ordered way and pattern of frequent items in the transaction is added to tree and nodes are connected to frequent items list and each other. This interconnection makes frequent pattern search faster avoiding the traversing of the entire tree. When considering the branch to be added for a transaction, the count of each node along a common prefix is incremented by 1. Nodes of same items are interconnected where most left one is connected to item in frequent items list. If the prefix of branch to be added does not exists then it is added as a new branch to root. After constructing the tree the mining proceeds as follows. Start from each frequent length-1 frequent item, construct its conditional pattern base, then construct its conditional tree and perform mining recursively on such a tree. The support of a candidate itemset is counted traversing the tree. The sum of count values at least frequent items nodes gives the support value.

**Algorithm 1:**

**Input:** Database of transaction and the minimum support count threshold

**Output**: List of frequent itemsets

```
Procedure Algorithm_1(Tree, A)
1. If Tree contains single path T then
2. For each combination (denoted as B) of the nodes in the path T
3. Generate pattern (B, A) appears together with minimum support of nodes in B;
4. Else for each $X_i$ in the header of Tree
5. Generate pattern B=($X_i$, A) with support=$X_i$. Support;
6. Construct B conditional pattern base and then B's conditional tree $Tree_B$;
7. If $Tree_B$!= 0 then
8. Call Algorithm_1 ($Tree_B$, B);
```

### 4.3. Algorithm 2:

Algorithm 2 is depth first search based search strategy, which adopts the technologies of vertical data format, lattice theory, equivalence classes, and intersection and so on. Algorithm 2 determines the support of an item set by constructing the list of identifiers of transactions that contain the item set. It does so by intersecting two lists of transaction identifiers of two item sets that differ only by one item and together form the item set currently processed. The main steps of Algorithm 2 are listed as follows: scan the database to get all frequent 1-itemsets, generate candidate 2-itemsets from frequent 1-itemsets, then get all frequent 2-itemsets by clipping non-frequent candidate itemsets; generate candidate 3-itemsets from frequent 2-itemsets and then get all frequent 3-itemsets by clipping non-frequent candidate itemsets; repeat the above steps, until no candidate itemset can be generated. Algorithm 2 adopts the join operation to generate candidate (K+1) itemset by taking the union of two k-itemset. The condition of two k-itemset can be joined is that the front k-1 items of the two k-itemset must be the same. Algorithm 2 divides the search space into multiple nonoverlapping sub spaces. The itemsets which have same prefix can be classified into a same class, and the generation of candidate itemsets can be only operated in a same class. The technology of equivalence classes can obviously improve the efficiency of generating candidate itemset and can reduce the occupation of memory.

**Algorithm 2:**

**Input**: Database of transaction and the minimum support count threshold

**Output:** List of frequent itemsets

1. Scan the database.

2. Transform the horizontally formatted data to the vertical format.

2. The frequent k-itemsets can be used to construct the candidate (k+1) itemsets based on the Apriori property.

3. This process repeats, with k incremented by 1 each time, until no frequent items or no candidate itemsets can be found.

### 5. Results and Analysis:

The analysis of results is very important and some assessment of their significance and quality must be given. Likely sources of error and inaccuracy should be mentioned. Use graphs, bar charts and histograms where appropriate, remembering to label all axes and give scales. Analysis provides an objective evaluation/comparison of the work with others. Analyze data and then prepare analyzed data in the form of graph, table or in the text form. A data analysis is a comprehensive summary of the results of your project and lists the main conclusions drawn from your tests and experiments. The analysis typically summarizes the qualitative and qualitative analysis that explains why some types of results are relevant. It includes comparison of algorithm with other algorithms. In below graph shows the difference between the timing of all algorithm while generating frequent set with same support. FDM[3] based algorithm takes more time for generate frequent itemset because before generate frequent itemset its generate candidate sets. Algorithm 1 takes more time because its first generate tree and then mined frequent itemset. Algorithm 2 takes more time because its convert horizontal transactional database to vertical transactional database and then mined frequent itemset and its take less time for execution.

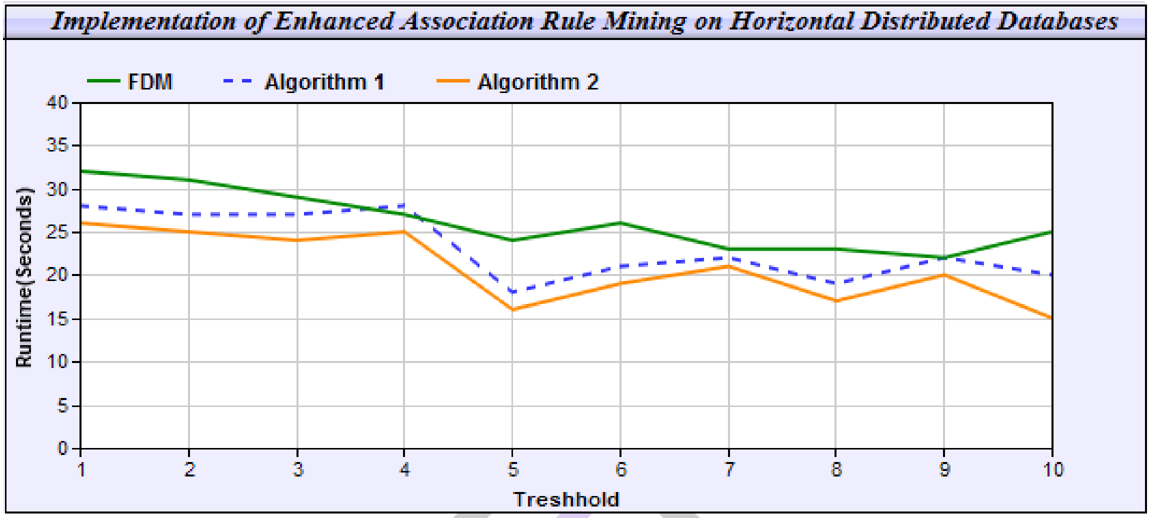

**Fig. 5.1 Comparison between Time to Generate Frequent Itemset of Different Algorithms with Respect to Support**

### 6. Conclusion and Future Work:

Implementation of Enhanced Association Rule Mining on Horizontal Distributed Databases is the application of data mining techniques to discover usage patterns from data, in order to understand and better serve the needs of applications and to cluster products around meaningful purchase opportunities related to use association. The algorithm used in present research generates association rules that associate the usage pattern of the clients for a particular data. Proposed algorithms for enhanced association rule mining on horizontal distributed databases that improves significantly upon the current leading protocol in terms of efficiency and privacy.

Proposed algorithms is applicable for very large databases where the available memory space is valuable and requires optimization. By Combining different algorithm that used for association rule mining it can be further tuned for better performance and efficiency. It can be applied in applications that deal in mining on live data on daily timely basis such as stock markets, financial statistics collection, weather forecasting etc. Its application can be utilized for industrial usage where precise pattern study is required with a large data sets to work on and so it can be modified according to their requirements.

### 7. REFERENCES:

[1] Tamir Tassa, "Secure mining of association rule in horizontally distributed databases", IEEE trans. Konwledege and Data Engg.,Vol. 26, no.2, April 2014.
[2] M. Kantarcioglu and C. Clifton, "Privacy-Preserving Distributed Mining of Association Rules on Horizontally Partitioned Data," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 9, pp. 1026-1037.
[3] D.W.L. Cheung, J. Han, V.T.Y. Ng, A.W.C. Fu, and Y. Fu. "A fast distributed algorithm for mining association rules". In PDIS, pages 31–42.
[4]Agrawal.R and Srikant.R. "Fast algorithms for mining association rules". In Proc. Int'l Conf. Very Large Data Bases (VLDB), pages 487–499, Sept. 1994.
[5] Ioannidis and Grama and Atallah. "A secure protocol for computing dot-products in clustered and distributed environments". In,pages 379–384.
[6] Ghosh and B. Nath. Multi-objective rule mining using genetic algorithm.pages no 123–133.
[7] J. Vaidya and C. Clifton, ―Privacy preserving association rule mining in vertically partitioned data, in The Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, July 23-26 2002, pp. 639–644.
[8] M. Freedman, Y. Ishai, B. Pinkas, and O. Reingold., ―Keyword search and oblivious pseudorandom functions‖, In *TCC*, pages 303–324, 2005.