International Journal of Advanced Science and Technology
Vol.102 (2017), pp.49-60
http://dx.doi.org/10.14257/ijast.2017.102.05

Comparative Analysis of Association Rule Mining Algorithms for the Distributed Data

K. S. Ranjith, Yang Zhenning, Ronnie D. Caytiles* and N. Ch. S. N. Iyengar

SCOPE, Vellore Institute of Technology University, Vellore-632014, TN, India
**Department of Multimedia Engineering, Hannam University, Korea,*
ksranjith2000@gmail.com, yang.zhenning2015@vit.ac.in rdcaytiles@gmail.com,
nchsniyr@vit.ac.in

Abstract

Many current data mining tasks can be accomplished successfully only in a distributed setting. The field of distributed data mining has therefore gained increasing importance in the last decade. The Apriori algorithm by Rakesh Agarwal has emerged as one of the best Association Rule mining algorithms. FP Growth also serves as the base algorithm for most parallel algorithms. The enormity and high dimensionality of datasets typically available as input to problem of association rule discovery, makes it an ideal problem for solving on multiple processors in parallel. The primary reasons are the memory and CPU speed limitations faced by single processors. In this paper an Association Rule mining algorithms for geographically distributed data is used in parallel and distributed environment so that it reduces communication costs. The response time is calculated in this environment using Supermarket data.

***Keywords**: Data mining, Apriori Algorithm, FP Growth Algorithm, Association Rules Mining*

1. Introduction

Association rule mining (ARM) has become one of the core data mining tasks and has attracted tremendous interest among data mining researchers. ARM is an undirected or unsupervised data mining technique which works on variable length data, and produces clear and understandable results. There are two dominant approaches for utilizing multiple Processors that have emerged; distributed memory in which each processor has a private memory; and shared memory in which all processors access common memory [5]. Shared memory architecture has many desirable properties. Each processor has direct and equal access to all memory in the system. Parallel programs are easy to implement On such a system. In distributed memory architecture each processor has its own local memory that can only be accessed directly by that processor [10]. For a processor to have access to data in the local memory of another processor a copy of the desired data element must be sent from one processor to the other through message passing. CSV data are used with the Optimized Distributed Association Rule Mining Algorithm.

The majority of the recognized organizations have accumulated masses of information from their customers for decades. With the e-commerce applications growing quickly, the organizations will have a vast quantity of data in months not in years. Data Mining, also called as Knowledge Discovery in Databases (KDD), is to determine trends, patterns, correlations, anomalies in these databases that can assist to create precise future decisions. Mining Association Rules is one of the most important application fields of Data Mining. Provided a set of customer transactions on items, the main intention is to determine correlations among the sales of items. Mining association rules, also known as market basket analysis, is one of the application fields of Data Mining. Think a market with a gathering of large amount of customer transactions. An association rule is X⇒Y, where X is referred as the antecedent and Y is referred as the consequent. X and Y are sets of items and the rule represents those customers who purchase X probable to purchase Y with probability %c where c is known as the confidence. Such a rule may be: “Eighty percent of people who purchase cigarettes also purchase matches". Such rules assists to respond questions of the variety “What is Coca Cola sold with?" or if the users are intended in checking the dependency among two items A and B it is required to determine rules that have A in the consequent and B in the antecedent. Here used a typical Market basket analysis. This is a perfect example for illustrating association rule mining. It is a fact that all the managers in any kind of shop or departmental stores would like to gain knowledge about the buying behaviour of every customers. This market basket analysis system will help the managers to understand about the sets of items are customers likely to purchase. This analysis may be carried out on all the retail stores data of customer transactions. These results will guide them to plan marketing or advertising approach. For example, market basket analysis will also help managers to propose new way of arrangement in store layouts. Based on this analysis, items that are regularly purchased together can be placed in close proximity with the purpose of further promote the sale of such items together. If consumers who purchase computers also likely to purchase anti-virus software at the same time, then placing the hardware display close to the software display will help to enhance the sales of both of these items.

Classification rule mining intends to determine a small set of regulations in the database that forms a perfect classifier. Association rule mining discovers all the rules offered in the database that assures some minimum support and minimum confidence constraint. In the case of association rule mining, the goal of discovery is not pre-determined, while for classification rule mining there is only one predetermined goal.

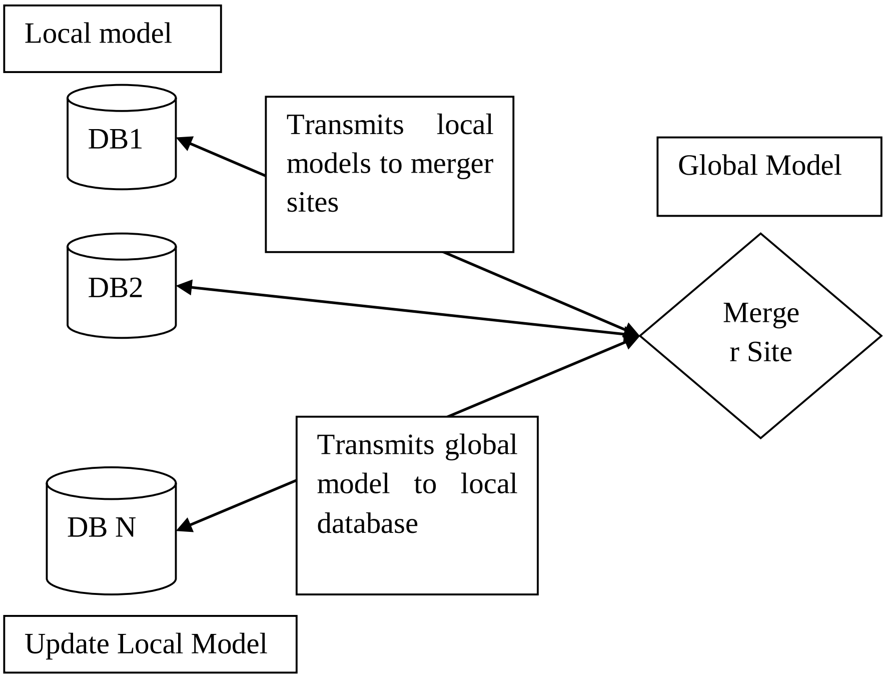

Local model
DB1
Transmits local models to merger sites
Global Model
DB2
Merge r Site
Transmits global model to local database
DB N
Update Local Model

Figure 1. Typical Architecture of Distributed Database Approach

The main aim of data mining technology is to explore hidden information from large databases. Many data mining techniques are exist such as association rule mining, clustering, classification and so on are well known and have wide applications in the real world [1-3]. The issue of privacy arises in two situations namely centralized and distributed environment. In centralized environment, database is available in single location and the multiple users are allowed to access the database. The main aim of privacy preserving data mining in this situation is to perform the mining process by hiding sensitive data/information from users. In distributed environment [4, 5], the database is available across multiple sites and the main aim of privacy preserving data mining in this environment is to find the global mining results by preserving the individual sites private data/information. Every site can access the global results which are useful for analysis

A. Database

A collection of related data, information and related pieces of data representing /capturing [6] [7] the information about a real-world enterprise or part of an enterprise. An Example University Database: Data about students, faculty, courses, research-laboratories, course registration/enrolment *etc*.

B. Distributed Database

A distributed database is a database in which storage devices are not all attached to a common processing unit such as the CPU, controlled by a distributed database management system [8] [9] (together sometimes called a distributed database system). In distributed database environment, the database among different sites can be partitioned as horizontally, vertically and mixed mode

A Parallel application could be divided into number of tasks and executed concurrently on different processors in the system [9]. However, the performance of a parallel application on a distributed system is mainly dependent on the allocation of the tasks comprising the application onto the available processors in the system. In different kinds of information databases, such as scientific data, medical data, financial data, and marketing transaction data; analysis and finding critical hidden information has been a focused area for researchers of data mining. How to effectively analyze and apply these data and find the critical hidden information from these databases, data mining technique has been the most widely discussed and frequently applied tool from recent decades. Although the data mining has been successfully applied in the areas of scientific analysis, business application, and medical research and its computational efficiency and accuracy are also improving, still manual works are required to complete the process of extraction. Association rule mining model among data mining several models, including Association rules, clustering and classification models, is the most widely applied method. The Apriori algorithm is the most representative algorithm for association rule mining. It consists of many modified algorithms that focus on improving its efficiency and accuracy. For the purpose of simulation, I have employed the database of Industries to assess the proposed algorithm. The rest of this paper is organized as follows. Section 2 briefly presents the general background, while the proposed method is explained in Section 3. Sections 4 illustrate the computational results of the Super market database. The concluding remarks are finally made in Section 5.

2. Literature Survey

Association Rule Mining: In data mining, association rule Learning is a popular and well researched method for discovering interesting relations between variables in large databases. It analyzes and present strong rules discovered in databases using different measures of interestingness. Based on the concept of Strong, rules, Agrawal *et al.*, introduced association rules for discovering regularities between products in large scale transaction data recorded by point-of-sale (POS) systems in supermarkets. For example, the rule found in the sales data of a supermarket would indicate that if a customer buys onions and potatoes together, he or she is likely to also buy burger. Such information can be used as the basis for decisions about marketing activities such as, *e.g.*, promotional pricing or product placements. In addition to the above example from market basket analysis association rules are employed today in many application areas including Web usage mining, intrusion detection and bioinformatics. Three parallel algorithms for mining association rules [3], an important data mining problem is formulated in this paper.

Agrawal and Shafer (1996) discuss three parallel algorithms for mining association rules. One of those, the Count Distribution (CD) algorithm, focuses on minimizing the communication cost, and is therefore suitable for mining association rules in a distributed computing environment. CD uses the Apriori algorithm (Agrawal and Srikant, 1994) locally at each data site. In each pass *k* of the algorithm, each site generates the same candidate *k*-itemsets based on the globally frequent itemsets of the previous phase. Then, each site calculates the local support counts of the candidate itemsets and broadcasts them to the rest of the sites, so that global support counts can be computed at each site. Subsequently, each site computes the *k*-frequent itemsets based on the global counts of the candidate itemsets. The communication complexity of CD in pass *k* is $O(|Ck|n2)$, where *Ck* is the set of candidate *k*-itemsets and *n* is the number of sites. In addition, CD involves a synchronization step when each site waits to receive the local support counts from every other site. Another algorithm that is based on Apriori is the Distributed Mining of Association rules (DMA) algorithm (Cheung, Ng, Fu & Fu, 1996), which is also found as Fast Distributed Mining of association rules (FDM) algorithm in (Cheung, Han, Ng, Fu & Fu, 1996). DMA generates a smaller number of candidate itemsets than CD, by pruning at each site the itemsets that are not locally frequent. In addition, it uses polling sites to optimize the exchange of support counts among sites, reducing the communication complexity in pass k to O(|Ck|n), where Ck is the set of candidate k-itemsets and *n* is the number of sites. However, the performance enhancements of DMA over CD are based on the assumption that the data distributions at the different sites are skewed. When this assumption is violated, DMA actually introduces a larger overhead than CD due to its higher complexity. The Optimized Distributed Association rule Mining (ODAM) algorithm (Ashrafi, Taniar & Smith, 2004) follows the paradigm of CD and DMA, but attempts to minimize communication and synchronization costs in two ways.

At the local mining level, it proposes a technical extension to the Apriori algorithm. It reduces the size of transactions by: i)deleting the items that weren’t found frequent in the previous step and ii) deleting duplicate transactions, but keeping track of them through a counter. It then attempts to fit the remaining transaction into main memory in order to avoid disk access costs. At the communication level, it minimizes the total message exchange by sending support counts of candidate itemsets to a single site, called receiver. The receiver broadcasts the globally frequent itemsets back to the distributed sites.

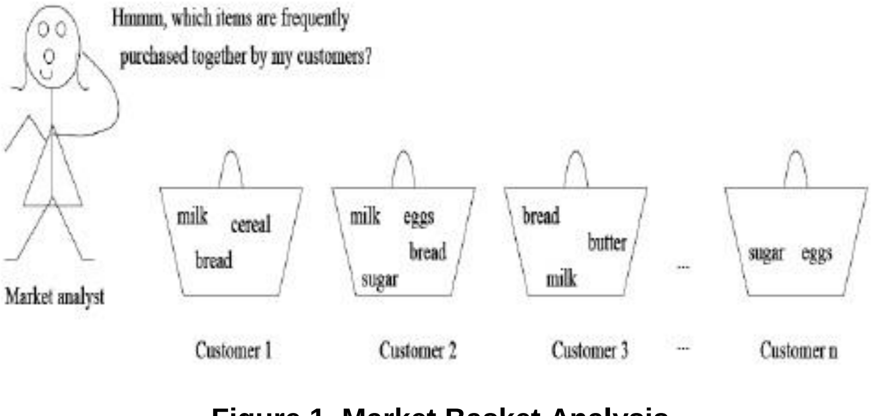


Figure 1. Market Basket Analysis

Zhixin *et al.*, [1] recommended an improved classification technique based on predictive association rules. Classification dependent predictive association rules (CPAR) is a type of association classification methods which unites the benefits of associative classification and conventional rule-based classification. For generation of the rule, CPAR is highly effective when compared to the conventional rule-based classification because most of the repeated calculation is ignored and multiple literals can be chosen to produce multiple rules at the same time. Even though the benefit mentioned above avoids the repeated calculation in rule generation, the prediction processes have the disadvantage in class rule distribution inconsistency and interruption of inaccurate class rules. Further, it is ineffective to instances satisfying no rules. To ignore these difficulties, the author recommends Class Weighting Adjustment, Centre Vector-based Pre-classification and Post-processing with Support Vector Machine (SVM). Experimental observations on Chinese text categorization corpus TanCorp proves that this approach gains an average enhancement of 5.91% on F1 score compared with CPAR. Qiang *et al.*, [2] presented association classification based method on compactness of rules. Associative classification provides maximum classification accurateness and strong flexibility [4]. On the other hand, this associative classification suffers from a difficulty of over fitting because the classification rules satisfied least support and lowest confidence are returned as strong association rules return to the classifier. In this paper, proposed an innovative association classification technique based on neatness of rules, it extends Apriori Algorithm which considers the interestingness, importance, overlapping relationships among rules. Experimental observation proves that the proposed approach has better classification accuracy in comparison with CBA and CMAR are highly intelligible. Wang *et al.*, [3] suggested a novel rule weighting approach in classification association rule mining. Classification association rule mining (CARM) is a latest classification rule mining technique that constructs an association rule mining based classifier by utilizing classification association rules (CARs).

In this paper, the alteration of the technique and required discretization of numeric characteristics are provided. Sumithra *et al.*, [5] proposed a distributed Apriori association rule and classical Apriori mining algorithms for grid based knowledge discovery. The intention of this paper is to obtain knowledge with the help of predictive Apriori and distributed grid dependent Apriori algorithms for association rule mining. The author provides the implementation of an association rules discovery data mining task with the help of Grid technologies. The author also provides a consequence of implementation with a contrast of existing Apriori and distributed Apriori. Distributed data mining systems offers an effective utilization of multiple processors and databases to accelerate the execution of data mining and facilitate data distribution. For evaluating the effectiveness of the described technique, performance investigation of Apriori and predictive Apriori techniques on a standard database have been provided using Weka tool [6]. Only little portions of the created rules would be of interest to several provided user. Therefore, numerous measures like confidence, support, lift, information gain, etc., have been suggested to find the best or highly interesting rules. On the other hand, some techniques are good at creating rules high in one interestingness measure but not good in other interestingness [7] measures. The relationship among the techniques and interestingness measures of the created rules is not clear until now. The author studied the relationship among the techniques and interesting measures. The author used synthetic data so that the outcome result is not restricted to particular situations.

Market Basket Analysis Based on Text Segmentation and Association Rule Mining is suggested by Xie *et al.*, [8]. Market basket analysis is very useful in offering scientific decision support for trade market by mining association rules between items people purchased collectively. The author provides an innovative market basket analysis technique by mining association rules on the items' internal features that are obtained with the help of automatic words segmentation technique. This technique has been used for dynamic dishes recommend system and results better in the experimental results. Chiu *et al.*, [9] proposed a market-basket analysis with principal component. Market-basket analysis is a well-known business crisis that can be solved computationally with the help of association rules, mined from transaction data to reduce the cross selling results[10]. The author model the market-basket analysis as a finite mixture density of human consumption activities based on social and cultural activities. The suggested technique is utilized to mine association rules to the 2002 student score list of computer dedicated field in Inner Mongolia university[12] of science and technology.

Yong *et al.*, [13] proposed a mining association rules with new measure criteria. In recent days, association rules mining from bulk databases is an active research field of data mining motivated by many application areas. But, there are some difficulties in the strong association rules mining depending on support confidence framework. Initially, there are a huge number of redundant association rules are created, then it is complicated for user to discover the interesting ones. Then, the correlation among the features of specified application areas is avoided. Therefore, the number of patterns itemsets reduced and it is effortless for user to gather the highly noticeable association rules. The simulation results suggest that the Chi-Squared test is efficient on decreasing the quantity of patterns through merging support and cover constrain. Pattern choosing according to Chi-Squared[11] test can remove few irrelevant attributes and the efficiency and veracity of mining association rules are enhanced. Mining traditional association rules using frequent itemsets lattice is given by Vo *et al.*, [14]. Numerous methods have been formulated for the enhancement of time in mining frequent itemsets. However, the methods which deal with the time of mining association rules were not put in deep research. In reality, in case of database which contains many frequent itemsets (from ten thousands up to millions), the time of mining association rules is much larger than that needed for mining frequent itemsets.

These algorithms have been designed to investigate and understand the performance implications of a spectrum of trade-offs between computation, communication, memory usage, synchronization, and the use of problem-specific information in parallel data mining [11]. Fast Distributed Mining of association rules, which generates a small number of candidate sets and substantially reduces the number of messages to be passed at mining association rules [4]. Algorithms for mining association rules from relational data have been well developed. Several query languages have been proposed, to assist association rule mining such as [12], [13]. The topic of mining XML data has received little attention, as the data mining community has focused on the development of techniques for extracting common structure from heterogeneous XML data. For instance, [14] has proposed an algorithm to construct a frequent tree by finding common sub trees embedded in the heterogeneous XML data. On the other hand, some researchers focus on developing a standard model to represent the knowledge extracted from the data using XML. JAM [15] has been developed to gather information from sparse data sources and induce a global classification model. The PADMA system [16] is a document analysis tool working on a distributed environment, based on cooperative agents. It works without any relational database underneath. Instead, there are PADMA agents that perform several relational operations with the information extracted from the documents.

3. Proposed Methods

The existing approaches of the Apriori and FP Growth Algorithm are used for the distributed dataset called Supermarket.

3.1 Apriori Algorithm

General Process

Association rule generation is usually split up into two separate steps:

1. First, minimum support is applied to find all frequent item sets in a database.
2. Second, these frequent item sets and the minimum confidence constraint are used to form rules.

Apriori Algorithm Pseudo code

```
Step.1 L1 = find frequent 1-itemsets(D);
Step.2 for $k = 2; L_{k-1} \neq \varnothing; k++$
            $C_k$ = apriori gen($L_{k-1}$);
            for each transaction t $\in$ D  // scan D for counts
            $C_t$ = subset($C_k$, t); // get the subsets of t that are candidates
                  for each candidate c $\in$ $C_t$; c.count++;
            $L_k$ = {c $\in$ $C_k$|c.count $\geq$ min_sup}
             return L = $U_k L_k$;
```

procedure Apriori gen(L_{k-1}:frequent (k -1)-itemsets)

```
Step.1 for each itemset $l_1 \in L_{k-1}$
            for each itemset $l_2 \in L_{k-1}$
            if $(l_1[1] = l_2[1] \wedge l_1[2] = l_2[2]) \wedge ... \wedge (l_1[k ..2] = l_2[k ..2]) \wedge (l_1[k ..1] < l_2[k ..1])$
            then  $c = l_1 \otimes l_2$;
      //generate candidate set joint step
Step 2. if has infrequent subset(c, $L_{k-1}$) then
      delete c; // prune step: remove unfruitful candidate
Step 3. else add c to $C_k$;
Step 4. return $C_k$;
```

procedure has infrequent subset(c: candidate k-itemset; L_{k-1}: frequent (k -1)-itemsets); use prior knowledge

```
Step 1. for each (k -1)-subset s of c
            if s $\notin$ $L_{k-1}$ then
Step 2. return TRUE;
Step 3. else return FALSE;
```

3.2 FP Growth Algorithm

Algorithm 1 (FP-tree construction)

Step 1. Scan the transaction database DB once. Collect the set of frequent items F and their supports. Sort F in support descending order as L, the list of frequent items.

Step 2. Create the root of an FP-tree, T, and label it as "Null". For each transaction, Trans in DB do the following.

Step 3. Select and sort the frequent items in Trans according to the order of L. Let the sorted frequent item list in Trans be [p|P], where p is the first element and P is the remaining list.

Step 4. Call insert tree([p|P], T),

The function insert tree([p|P]; T) is performed as follows. If T has a child N such that N.item-name= p.item-name, then increment N's count by 1;

Step 5. else create a new node N, and let its count be 1, its parent link is linked to T, and its node link be linked to the nodes with the same item-name via the node-link structure. If P is nonempty, call insert tree(P;N) recursively.

```
Algorithm 2 (Mining Frequent Itemsets)
Step 1.  Call FP-growth (FP-tree ; null).
Step 2. if Tree contains a single path then for each combination (denoted as β)
of the nodes in the path P do
Step 3. generate pattern $\beta \cup \alpha$ with support = minimum support of nodes in β ;
Step 4. else for each $a_i$ in the header of Tree do generate pattern $\beta = a_i \cup \alpha$ with
support = $a_i$.support;
Step 5. construct β 's conditional pattern base and then β 's conditional FP-tree $Tree_\beta$;
Step 6.  if $Tree_\beta \neq \alpha$ then call FP-growth ($Tree_\beta$, β)
```

3.3 Distributed Mining of Association Rules(DMA)

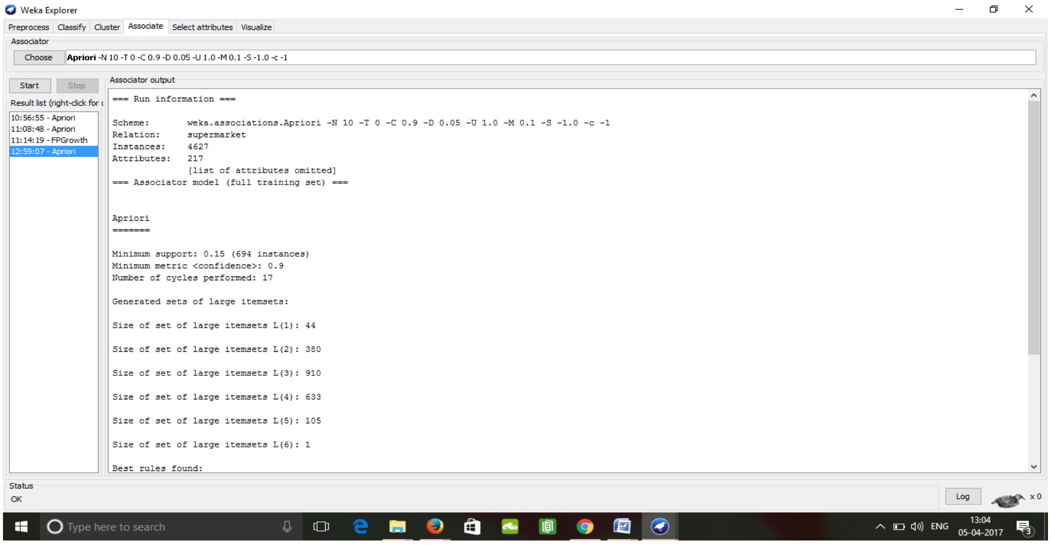
```
Algorithm: Distributed Mining of Association rules algorithm
Input: 1) DB'. the database partition at each site, (its size is equal to $D^i$)
        2) s: the minimum support threshold; both submitted at each site $S^i$, (i=1,...n);
Output: L: the set of all large item sets in DB, returned at every site;
Method: iterates the following program fragment distributive at each site $S^i$ starting from
k = 1, where k is the iteration loop counter; the algorithm terminates when either $L_k$,
returned is empty or the set of candidate sets $CH_k$ is empty
Step 1: /* Local Pruning */
        if k = 1 then scan $DB^i$ to compute $T_1^i$; / * $T_1^i$ is an array containing all size 1 -
        itemsets in $DB^i$ their local support counts in site $S^i$ */
Step 2: else {
        C $H_k$ = $\bigcup_{i=1}^{n} CH_k^i$ =$\bigcup_{i=1}^{n}$ Apriori-gen ( $HL_{k-1}^i$ ); / * generate size k - candidate
sets */
        scan $DB^i$ to built the hash tree $T_k^i$ ; } /* $T_k^i$ contains all candidate sets in $CH_k$ and
        their support counts in site $S^i$ */
Step 3: for_all X ∈ $T_k^i$ do if X.$sup^i$ ≥ s x $D^i$ then
Step 4: for j=1 to n do for-all X ∈ $LL_k^{i,j}$ do
        if polling_site(X) = $S^j$ then add (X, X.$sup^i$) in to $LL_k^{i,j}$
Step: 5: /* compute the locally large candidates and divide them according to their
polling
                sites , end Candidates to Polling Sites */
        for j=1, ..., n do send $LL_k^{i,j}$ to site $S^j$;
Step 6: /*Receive Candidates as a Polling Site*/
        for j = 1, . . ., n do { receive $LL_k^{i,j}$ ;
        for_all X ∈ $LL_k^{j,i}$ do { store X in $LP_k^i$ ;
        update X.lauge-sites in $LP_k^i$ to record the sites at which X is locally large; }
        }
Step 7: /*Send Polling Requests as a Polling Site to collect support counts*/
        for_all X ∈ $LP_k^i$ do { broadcast polling requests for X to the sites $S^j$, where
        $S^j$ ∉ X.large_sites; receive X.$sup^j$ from the sites $S^j$, where $S^j$ ∉ X.large_sites; };
Step 8: /*Compute Global Support Counts and Heavy itemsets*/
        for_all X ∈ $LP_k^i$ do { X.sup = $\sum_{i=1}^{n} X.\mathrm{sup}^i$
        if X.sup ≥ s x D then insert X into $H_k^i$ ; /* filter out the heavy k-itemsets;*/ };
Step 9: broadcast $H_k^i$ ; receive $H_k^j$ from all other sites $S^j$, (j ≠ i);
```

return $L_k = \bigcup_{i=1}^{n} H_k^i$

4. Analytical Results

The results are performed by using the WEKA tool. First the association rules for Supermarket dataset by using both the Apriori and FP Growth Algorithms has been generated separately. Then both the algorithms with Supermarket Dataset are compared by the parameter (Execution Time) speed of generating the frequent patterns and the graph is generated based on the results.

The Apriori algorithm generates the large frequent item sets in a database. The property of the Apriori algorithms is the all item sets in a large frequent item set is also frequent. So it generates only the large frequent item sets.

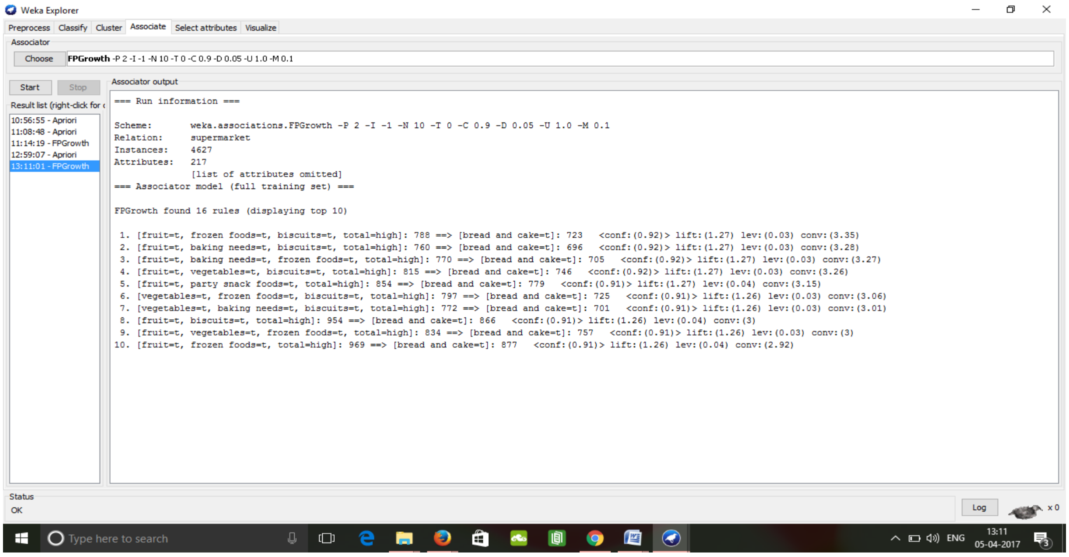


Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1

10:56:55 - Apriori
11:08:48 - Apriori
11:14:19 - FPGrowth
12:59:07 - Apriori

```
=== Run information ===

Scheme:       weka.associations.Apriori -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1
Relation:     supermarket
Instances:    4627
Attributes:   217
              [list of attributes omitted]
=== Associator model (full training set) ===


Apriori
=======

Minimum support: 0.15 (694 instances)
Minimum metric <confidence>: 0.9
Number of cycles performed: 17

Generated sets of large itemsets:

Size of set of large itemsets L(1): 44

Size of set of large itemsets L(2): 380

Size of set of large itemsets L(3): 910

Size of set of large itemsets L(4): 633

Size of set of large itemsets L(5): 105

Size of set of large itemsets L(6): 1

Best rules found:
```

Figure 3.1. Generation of Association Rules by Apriori Algorithm



FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1

10:56:55 - Apriori
11:08:48 - Apriori
11:14:19 - FPGrowth
12:59:07 - Apriori
13:11:01 - FPGrowth

```
=== Run information ===

Scheme:       weka.associations.FPGrowth -P 2 -I -1 -N 10 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.1
Relation:     supermarket
Instances:    4627
Attributes:   217
              [list of attributes omitted]
=== Associator model (full training set) ===

FPGrowth found 16 rules (displaying top 10)

 1. [fruit=t, frozen foods=t, biscuits=t, total=high]: 788 ==> [bread and cake=t]: 723   <conf:(0.92)> lift:(1.27) lev:(0.03) conv:(3.35)
 2. [fruit=t, baking needs=t, biscuits=t, total=high]: 760 ==> [bread and cake=t]: 696   <conf:(0.92)> lift:(1.27) lev:(0.03) conv:(3.28)
 3. [fruit=t, baking needs=t, frozen foods=t, total=high]: 770 ==> [bread and cake=t]: 705   <conf:(0.92)> lift:(1.27) lev:(0.03) conv:(3.27)
 4. [fruit=t, vegetables=t, biscuits=t, total=high]: 815 ==> [bread and cake=t]: 746   <conf:(0.92)> lift:(1.27) lev:(0.03) conv:(3.26)
 5. [fruit=t, party snack foods=t, total=high]: 854 ==> [bread and cake=t]: 779   <conf:(0.91)> lift:(1.27) lev:(0.04) conv:(3.15)
 6. [vegetables=t, frozen foods=t, biscuits=t, total=high]: 797 ==> [bread and cake=t]: 725   <conf:(0.91)> lift:(1.26) lev:(0.03) conv:(3.06)
 7. [vegetables=t, baking needs=t, biscuits=t, total=high]: 772 ==> [bread and cake=t]: 701   <conf:(0.91)> lift:(1.26) lev:(0.03) conv:(3.01)
 8. [fruit=t, biscuits=t, total=high]: 954 ==> [bread and cake=t]: 866   <conf:(0.91)> lift:(1.26) lev:(0.04) conv:(3)
 9. [fruit=t, vegetables=t, frozen foods=t, total=high]: 834 ==> [bread and cake=t]: 757   <conf:(0.91)> lift:(1.26) lev:(0.03) conv:(3)
10. [fruit=t, frozen foods=t, total=high]: 969 ==> [bread and cake=t]: 877   <conf:(0.91)> lift:(1.26) lev:(0.04) conv:(2.92)
```

Figure 3.2. Generation of Association Rules by FP Growth Algorithm

The Distributed Mining of Association rule Algorithm is the Modified algorithm of the FP Growth algorithm, where it generates the best association rules in less time compare to the Apriori and FP Growth Algorithm. The Comparison of the algorithms graph is plotted based upon the values obtained in weka tool. Clearly the graph shows that the DMA algorithm is best for mining association rules in distributed databases.

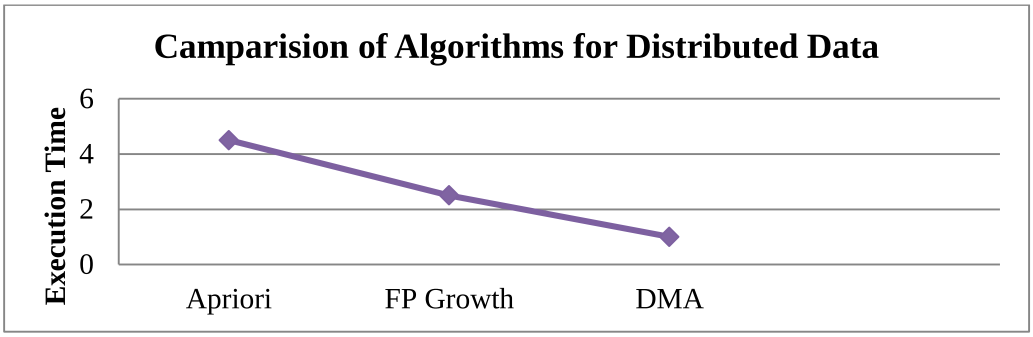


Figure 3.3. Comparisons of Algorithms in terms of Execution Time

5. Conclusion

Data mining is an important area where the use cases will exist almost in every field. Mining Association Rules is major research area in data mining. The Apriori and FP Growth Algorithms are base algorithms for many mining association rule algorithms, This paper presents the generation of Association rules by using the weka tool for the Apriori and FP Growth algorithm, comparison between the Apriori and FP growth Algorithms, and proved that the FP Growth algorithms is fast in execution compared to the Apriori Algorithms.

References

[1] S. 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, vol. 2, no. 2, **(2010)**, pp. 10-23.

[2] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules in Large Database," 2nd Conf. Very Large Databases (VLDB 94), **(1994)** Morgan.

[3] R. Agrawal and J. C. Shafer, “Parallel Mining of Association Rules”, IEEE Transactions on knowledge, and Data Engineering Distributed Systems, vol. 8, no. 6, **(2004)**, pp. 962-969.

[4] Cheung DW, Han J, Ng VT, Fu Aw and Fu Y., “A fast distributed algorithm for mining association rules”, In Parallel and Distributed Information Systems, 1996., Fourth International Conference, Chilli, **(1996)** December18, pp. 31-42.

[5] A. Savasere, E. R. Omiecinski and S. B. Navathe, “An efficient algorithm for mining association rules in large databases”, Georgia Institute of Technology, Argentina, **(1995)**, pp. 18-25.

[6] J. Han, J. Pei and Y. Yin, “Mining Frequent Patterns without Candidate Generation”, Int'l. Conf. Management of Data, ACM Press, **(2000)**, pp. 1-8.

[7] R. Agrawal, T. Imieliński and A. Swami, “Mining association rules between sets of items in large databases”, InAcm sigmod record, ACM, vol. 22, no. 2, **(1993)** June 1, pp. 207-216.

[8] F. Provost and V. Kolluri, “A survey of methods for scaling up inductive algorithms”, Data mining and knowledge discovery, vol. 3, no. 2, **(1999**) June 1, pp. 131-69.

[9] M. J. Zaki, “Parallel Data Mining for Association Rules on Shared -Memory Multiprocessors”, report TR,618, Computer Science Dept., Univ. of Rochester, **(1996)**.

[10] Cheung DW, Ng VT, Fu AW, and Fu Y, “Efficient mining of association rules in distributed databases” IEEE transactions on Knowledge and Data Engineering, vol. 8, no. 6, **(1996)** December, pp. 911-22.

[11] A. Schuster and R. Wolff, “Communication - Efficient Distributed Mining of Association Rules”, Proceedings of the ACM SIGMOD Int'l Conf. Management of Data, ACM Press, **(2001)**, pp. 473-484.

[12] T. Imieliński and A. Virmani, “MSQL: A query language for database mining”, Data Mining and Knowledge Discovery, vol. 3, no. 4, **(1999)** December 1, pp. 373-408.

[13] R. Meo, G. Psaila and S. Ceri, “A new SQL-like operator for mining association rules”, InVLDB **(1996)** September 3, vol. 9, no. 6, pp. 122-133.

[14] A. Termier, M. C. Rousset, Sebag and M. Treefinder, “A first step towards xml data mining”, In Data Mining, 2002. ICDM 2003. Proceedings. IEEE International Conference, IEEE, **(2002)**, pp. 450-457.

[15] A. Prodromidis, P. Chan and S. Stolfo, “Chapter Meta learning in distributed data mining systems: Issues and approaches”, In AAAI/MIT Press, **(2000)**.

[16] H. Kargupta, I. Hamzaoglu and B. Stafford, “Scalable, Distributed Data Mining-An Agent Architecture” In KDD, **(1997)** August 14, pp. 211-214.

[17] A. Y. Zomaya, T. El-Ghazawi, and O. Frieder, “Parallel and distributed computing for data mining”, IEEE sConcurrency, vol. 7, no. 4, **(1999)** October 1, pp. 3-11.

Authors

K. S. Ranjith, currently doing his research at VIT University. His research area are Wireless Sensor Networks, Text Mining and Big Data Predictive Analytics.

Yang Zhenning, he is pursuing M.ScComputer Science at School of Computing Science and Engineering, VIT University, Vellore. His area of interests are Algorithm design an Pattern Recognition, operating Systems and cloud computing

Ronnie D. Caytiles, He had his Bachelor of Science in Computer Engineering- Western Institute of Technology, Iloilo City, Philippines, and Master of Science in Computer Science– Central Philippine University, Iloilo City, Philippines. He finished his Ph.D. in Multimedia Engineering, Hannam University, Daejeon, Korea. Currently, he serves as an Assistant Professor at Multimedia Engineering department, Hannam University, Daejeon, Korea. His research interests include Mobile Computing, Multimedia Communication, Information Technology Security, Ubiquitous Computing, Control and Automation.

N. Ch. S. N. Iyengar, He is a Professor of the School of Computer Sciences and Engineering at VIT University, Vellore, TN, India. His research interests include Distributed Computing, Information Security, Intelligent Computing, and Fluid Dynamics (Porous Media). He has had teaching and research experience with a good number of publications in reputed International Journals & Conferences. He chaired many International Conferences delivered Keynote lectures, served as PC Member/Reviewer. He is an Editorial Board member formany International Journals like Int. J. of Advances in Science and Technology, of SERSC, Cybernetics and Information Technologies (CIT)-Bulgaria, Egyptian Computer Science Journal-Egypt, IJConvC of Inderscience-China, IJCA (USA) etc., Also Editor in Chief for International Journal of Software Engineering and Applications(IJSEA) of AIRCC, Advances in Computer Science (ASC) of PPH, Guest editor for “Cloud Computing and Services” IJCNS.