Frequent Pattern Mining Algorithms: A Review

Sukhbir Singh, Dharmender Kumar

Abstract*—Mining frequent patterns is one of the most important concepts of data mining. Frequent pattern mining has been a highly concerned field of data mining for researcher for over two decades. Several algorithms have been developed for finding frequent itemsets from the databases. The efficiency of these algorithms is a major issue since a long time and has captured the interest of a large community of researchers. In Literature review it is found that great effort has been made in this area so far to development of efficient and scalable algorithms for frequent itemset mining in various types of databases due to their importance in various fields. In 1993, R. Agrawal and R. Srikant first proposed the most classical association rule mining algorithm named as Apriori algorithm. But Apriori has two major drawbacks: large number of candidate itemsets generation and large no of database scan. Like most of the association rule algorithms, first it discover minimal frequent itemsets, then it discover the maximal frequent itemsets by using these minimal frequent itemsets, so all approach of this type take large time to find maximal frequent itemsets and needed large number of database scan, also not suitable for the continuous changing database. To overcome these problems, extensive work have done by many researchers, by enhancement and modification on basic algorithms like Apriori algorithm, FP growth algorithm, Eclat algorithm, and MFI algorithm etc. Maximal frequent itemset (MFI) was proposed by Bayard in the year 1998. (MFI) used to find maximal frequent item. After that lots of improved approaches have been proposed to efficiently mining the maximal frequent pattern such as Mafia, GenMax Smart-Miner etc. The present paper provides an overview of various frequent pattern mining algorithms with the expectation that it would serve as a reference material for researchers in this field.

Keywords*—Apriori Algorithm, Association Rules, Boolean matrix, Data Mining, Frequent Itemset, Maximal Frequent Itemset (MFI), Maximal frequent itemset first (MFIF).

I. INTRODUCTION

Data mining is the process of discovering meaningful new and interesting correlation, patterns and trends by sifting through large amounts of data, by using pattern recognition technologies as well as statistical and mathematical technique [63]. Now a days Data mining has been widely used and unifies research in various fields such as computer science, networking and engineering, statistics, databases, machine learning and Artificial Intelligence etc. There are different techniques that also fit in this category including association rule mining, classification and clustering as well as regression [1]. Finding association rules is the core process of data mining and it is the most popular technique has been studied by many researchers.. It is mining for association rules in database of sales transactions between items which is important field of the research in dataset [2]. The benefits of these rules are detecting unknown relationships, producing results which can used as a basis for decision making and prediction. For finding frequent pattern and then generating association rules by using these pattern, APRIORI algorithm was proposed by R. Agrawal and R. Srikant[3]. Many improvements have been proposed to enhance the performance of Apriori algorithm [4] because of some limitation of Apriori algorithm. One of these was given by Changsheng Zhang and Jing Ruan by dataset reduction approach and by decreasing the I/O spending [5]. For cross selling strategies organization in retail industry and to increase the sales, they have applied this modified algorithm. Wanjun Yu, Xiaochun Wang proposed RAAT (Reduced Apriori Algorithm with Tag) [6], which use transaction tag method to improves the performance of Apriori algorithm and in pruning operation, it reduces generation of frequent itemset. Dongme Sun, Sheohue Teng has proposed a new approach based on forward and backward scan of database [7]. If it applied with certain satisfying conditions, it produces the frequent itemsets more efficiently. P-matrix algorithm [8] proposed by Xinxi Dai and Sixue Bai, and in comparison to Apriori algorithm, P-matrix is faster and more efficient. Zhi Lin, Guoming Sang, Mingyu Lu [9] for finding association rules, proposed vector operation based method. Two algorithms called as Maximal Frequent Item (MFI) [20] and Maximal frequent itemset first (MFIF) [10] have been proposed which are much efficient and more attractive than other frequent itemset mining algorithms due to following reason:-

- Faster to generate maximal frequent itemset.
- Less no of database scan.
- Quite simple, flexible and robust.

Manuscript Received on September 2014.

Sukhbir Singh, Master of Technology, Department of Computer Science and Engineering, GJUS & T, Hisar, Haryana, India.

Dharmender Kumar, Assoc. Prof., Department of Computer Science and Engineering, GJUS & T, Hisar, Haryana, India.

II. ASSOCIATION RULE

Association rule mining aims to discover the relationships and the patterns in a dataset by including two steps (1) finding all frequent itemsets and (2) generating association rules from those frequent itemsets. In a database the frequency of an itemset, is also referred to as the support count, which represent the number of transactions that contain that itemset. An itemset is called frequent itemset if its support count is greater than or equal to the minimum support threshold. An association rule is like as: X->Y where $\text{Support}\,(X \rightarrow Y) \;=\; P\,(XUY)$ and $\text{Confidence}(X \rightarrow Y) \;=\text{P}(Y/X) \;=\; P\,(XUY)/\text{support count}(X).$
Minimum support threshold (s) and minimum confidence threshold (c) are used to remove the uninteresting association rules. The association rules are interesting if and only if it has both the support and the confidence greater than or equal to these thresholds [2].

A. Apriori Algorithm

Apriori algorithm [3] has been proposed by R. Agrawal and R. Srikant for finding frequent itemsets. In the first round, the Apriori algorithm scans the database to determine L_1 (line 1). In the k^{th} round, where $k \geq 2$, the process of the Apriori algorithm can be divided into the following three

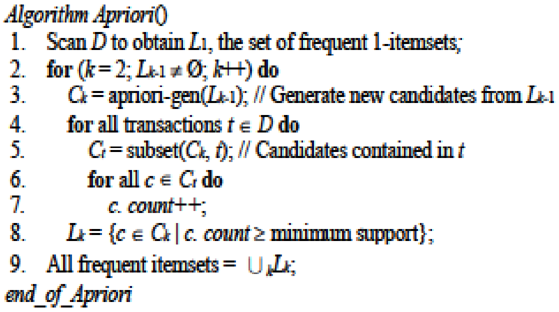
Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

steps.

Step 1. Line 3 constructs C_k from L_{k-1}, which was determined in the (k-1) th round.

Step 2. Lines 4-7 scan the database to count the support of each k-itemset in C_k.

Step 3. Line 9 determines the $L_{k,}$ whose support is greater than or equal to the user-specified minimum support, from C_k. Fig.1 shows the process of Apriori.



```
Algorithm Apriori()
1.  Scan $D$ to obtain $L_1$, the set of frequent 1-itemsets;
2.  for ($k = 2$; $L_{k-1} \neq \emptyset$; $k$++) do
3.      $C_k$ = apriori-gen($L_{k-1}$); // Generate new candidates from $L_{k-1}$
4.      for all transactions $t \in D$ do
5.          $C_t$ = subset($C_k$, $t$); // Candidates contained in $t$
6.          for all $c \in C_t$ do
7.              $c$. count++;
8.      $L_k = \{c \in C_k \mid c.\ count \geq$ minimum support$\}$;
9.  All frequent itemsets = $\cup_k L_k$;
end_of_Apriori
```

Fig. 1-Apriori Algorithm

The algorithm terminates when more candidate itemsets cannot be constructed for next round. The algorithm needs to do multiple database scans as many times as the length of the largest frequent itemset. Therefore, its performance decreases dramatically when the length of the largest frequent itemset is relatively long [11].

B. FP Growth Algorithm

The process of frequent patterns generation in FP-growth (frequent pattern growth) algorithm includes two sub processes: first is the construction of the FT-Tree, and 2nd is generating frequent patterns from the FP-Tree [12]. To store the database in a compressed form, it uses an extended prefix tree (FP-tree) structure. FP-growth uses a divide-and-conquer approach to decompose both the mining tasks and the databases. FP-Tree, recovers the two disadvantages of the Apriori, it takes only two scan of the database and no candidate generation. So FP-Tree is faster than the Apriori algorithm. It is more effective in dense databases than in sparse databases. Its major cost is the recursive construction of the FP-trees [13].

C. Partitioning Algorithm

To overcome the memory problem for large database which can not fit into main memory Partitioning algorithm is used to find the frequent elements. It is based on the partitioning of database in n parts [14], because small parts of database easily fit into main memory.

D. Direct Hashing and Pruning (DHP) Algorithm

A DHP technique use Hash table structure. It reduces the number of candidates in the early passes Ck for k > 1 and the size of database [15]. In DHP technique, support is counted by mapping the items from the candidate list into the buckets. In DHP technique, when a new itemset is occurred, it checks the itemset exist earlier or not, if exist it increases the bucket count else insert itemset into new bucket. And in the end the buckets which have less support count than the minimum support is deleted from the candidate set.

E. Sampling Algorithm

In Sampling algorithm, a random sample is picked up in such a way that the sample can be fit in the main memory, and frequent pattern are mining from this sample. This removes the I/O overhead by not taking the complete database but only a sample of database for checking the frequency [16].

F. Eclat

Eclat [17, 18] algorithm uses a depth-first approach with the set intersection, and vertical data format. Each item is stored together with its cover (also called tid list). The support count of an itemset X can be easily computed by intersecting the any two subsets of X , like Y and Z are subset of X, such that Y U Z = X.

G. The Pincer-Search Algorithm

For mining maximal frequent itemsets, Lin and Kedem [19] presented a new approach by combining both top-down and bottom-up approach; it reduces the complexity for generating maximal frequent itemsets. The bottom-up approach starts from 1-itemset, moves one-level up in each iteration and proceeds up to *n*-itemsets like Apriori algorithm while the top-down approach starts from n itemsets, moves many levels down in each iteration and proceeds up to 1-itemset. Both bottoms-up and top-down approach individually identify the maximal frequent itemsets by examining its candidates.

III. PREVIOUS WORKS ON FREQUENT PATTERN MINING ALGORITHM

Association rule mining, one of the most important and well researched techniques of data mining, was first introduced by Agarwal , R., Imielinski T., and Swami, A. N. in “Mining association rules between sets of items in large databases” [35]. Association rules are used in various fields such as telecommunication networks, online shopping, inventory control, marketing and risk management etc. Association rule mining is used to find out association rules that satisfy the user-defined minimum support threshold and confidence threshold from a given database. An efficient algorithm proposed by Agarwal, R. Aggarwal, C. and Prasad V., called as TreeProjection [36]. The general concept of Tree Projection is that it builds a lexicographical tree and on the base of frequent patterns mined so far, TreeProjection projects a large database into a set of reduced item-based sub-databases. The total no. of frequent itemsets is equal to the total no. of nodes in its lexicographic tree. Two main factors explained the efficiency of TreeProjection : (1st) the support counting in a small space by the transaction projection; and (2nd) the management and counting of candidates, providing the flexibility of picking efficient strategy during the tree generation and transaction projection is facilitated by the lexicographical tree. An efficient algorithm for mining association rules named as PRICES is proposed by Wang and Tjortjis [37] .The most time-consuming step of generation of large itemset is reduced by scanning the database only once and by using logical operations. Matrix Algorithm for generating large frequent candidate sets is proposed by Yuan, Y. And Huang, T. [38], Which generates a Boolean matrix by scanning the whole database only once, and the frequent candidate sets are generated from the generated matrix. And the generated frequent candidate sets used to mine association rules. The algorithm is better than Apriori Algorithm. A sampling approach for association rule mining is proposed by

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

Toivonen [39]. It has two steps: (1st) a random sample of the whole database is taken and all associations in the sample are mined. (2nd) these results are verified with the rest of database. To maximize the effectiveness, lower minimum support threshold is used on sample. Some associations which were not frequent in the sample but frequent in the whole database (main drawback of this approach) are used to construct the complete set of associations in the 2nd step. Chuang et al. [41] presented Sampling Error Estimation (SEE). It is a progressive sampling algorithm; it is used to determining an appropriate sample size for mining association rules. Sampling Error Estimation (SEE) has two advantages. (1st)- SEE is very efficient because an appropriate sample size can be determined, without the need of executing association rules. (2nd)- SEE is highly accurate to determine sample size, meaning that on this size of sample, to find a sufficiently accurate result, association rules can be very efficiently executed. For sampling large datasets with replacement, the sufficient sample size based on central limit theorem is derived by Li and Gopalan [42].Association rule mining approaches have been adapted the parallelism so that the advantage of the larger storage capacity and higher speed of parallel system can be taken [43]. FDM presented by Cheung et al. [44], which is Parallelization of Apriori algorithm, each machine with its own partition of the database and no sharing. The database scan is performed independently at every level and on each machine, on the local partition. Pruning approach is also distributed. Like FDM, candidates in DDM (D-ARM algorithm presented by Schuster and Wolff [45]) are generated level wise and then counted by each node in its local database. Then the nodes execute a distributed decision protocol in order to find out frequent candidates set. FPM (Fast Parallel Mining) for mining association rules on a shared-nothing parallel system has been presented by Cheung, D. And Xiao, Y. [46] is another efficient parallel algorithm. FPM uses the distribution count approach and use two efficient candidate pruning techniques, i.e., global and distributed pruning technique. It employs a simple communication technique; in each pass, only one round of message exchanges takes place. DAA (Data Allocation Algorithm) is proposed by Manning, A., Keane, J. [47]. DAA uses Principal Component Analysis to improve the data distribution. A recent survey on parallel association rule mining with shared memory architecture which cover most of the techniques adopted, trends and challenges in parallel data mining has been written by Parthasarathy, S., Zaki, M. J., Ogihara, M., [48]. All techniques are Apriori-based. A parallelization scheme which can be used to parallelize the fast and efficient frequent pattern mining algorithms based on FP-trees has been described by Tang and Turkia [49]. An approach Relim (Recursive elimination) processes the transactions directly, based on FP-Growth algorithm without the prefix tree. FP-Growth algorithm uses the prefix tree for representation of datasets, which save a lot of memory for storing the transaction. Relim algorithm deletes all items from the transaction database that has least frequent items. Relim is better when min support is low [40]. The goal of frequent pattern mining algorithm is discover all the patterns having support greater than the user-defined threshold. But, many time users want the set of patterns to be discovered according to some extra constraints applied by the user on the structure of patterns. Techniques applicable to constraint-driven pattern discovery or mining can be classified into the following three groups:

1. Post-processing Techniques (after the actual discovery process, remove the patterns that do not satisfy the user-specified Constraints);
2. Pattern filtering Techniques (to generate only those patterns which satisfy the constraints, add the pattern constraints into the actual mining process);
3. Database filtering Techniques (restrict the source database to objects that can contain the patterns that satisfy pattern constraints). By using database filtering approach, Wojciechowski and Zakrzewicz [50] focus on enhance the performance of constraint- based frequent pattern mining. Database filtering techniques restrict the source database to objects that can contain the patterns that satisfy pattern constraints and resulting database is small. Tien Dung Do [51] has been proposed a different type of constraints called as category-based as well as the associated algorithm for constrained rule mining based on Apriori algorithm. By bypassing most of the subsets of the final itemsets, the Category-based Apriori algorithm decreases the computational complexity of the mining process. Rapid Association Rule Mining (RARM) [52] is an association rule mining technique that avoids the candidate generation process and uses the tree structure to represent the original database. To enhance the performance of existing mining algorithms, some constraints were applied during the mining process to generate only interesting and useful association rules instead of all the association rules means optimization.

Since 1993, researcher has been made lots of improvements to the Apriori algorithm. Requirements for the improved algorithm in “One Optimized Method of Apriori Algorithm” [53], itemsets needed to be arrange lexicographically, and compression of Boolean matrix is not completed and still required a lot of space in the calculation process. The improved algorithm by WANG Chengliang and WU Yanjuan [54] only converts the database to a Boolean matrix, and calculate the frequent itemsets with the method of vector inner-product. There is no compression of Boolean matrix and nothing about the weight concept. Improved algorithm by ZENG Wandan, ZHOU Xubo, DAI Bo, CHANG Guiran and LI Chunping [55] and by ZHANG Yueqin [56] added a new column in the Boolean matrix, but only the row vector compression of matrix and no compression on the columns of matrix. The improved algorithm of “Algorithm for Generating Strong Association Rules Based on Matrix” [57] is a kind of algorithm based on the sort of matrix algorithm and this improved algorithm has certain advantages in generating frequent itemsets, but there isn't much improvement for data compression, and Boolean matrix sorting process also costly. Improved algorithm in [58] is generated by 2-itemsets support matrix, which avoids the invalid 2-itemsets and solves its bottleneck problem, but it still needs repeating the scanning of matrix during generating frequent itemsets, and the only solving efficiency of 2-itemsets is more obvious. As mentioned above, based on the current study, Zhiyong Wang [59] has made the Improvements in the Apriori algorithm. It compresses the row vector and column vector of Boolean matrix in two directions and it introduced the weight vector inner product and the algorithm of frequent itemsets. A new method is presented by Harpreet Singh and Renu Dhir [60], called as MBAT (Matrix Based Algorithm with Tags). MBAT generate transactional matrix from the database. And then mine the frequent itemsets directly from this transactional matrix by using the tags to count support of the itemsets. The number of candidate itemsets, mainly candidate 2-itemsets greatly decreases by proposed approach.

IV. VARIOUS MAXIMAL FREQUENT ITEMSET MINING ALGORITHM

Maximal Frequent Itemset (MFI) was first introduced in the year 1998, Bayardo proposed MaxMiner Algorithm [20] for mining the maximal itemsets. MaxMiner uses a breadth-first search, and also it decrease the no. of scans of database by using a look-ahead to prune the branches of the tree i.e., it involves in superset pruning. To enhance the effectiveness of superset frequency pruning, MaxMiner also uses dynamic re-ordering of items. It uses the horizontal dataset format so number of scan of database is equal to Apriori. Depth Project Algorithm [21], proposed by Agrawal, also uses the depth first search of a lexicographic tree and also uses counting method based on transaction projections along with its branches to find long itemsets. The DepthProject algorithm also perform a look-ahead pruning method with item re-ordering and returns a superset of the maximal frequent itemset and required a post-pruning to eliminate non-maximal itemsets. GenMax [22, 23] was proposed by Gouda and Zaki, backtracking approach used to identify all maximal itemsets. The data representation in GenMax is in vertical format. It also uses a progressive focussing approach to remove non-maximal itemsets and uses Diffset propagation to perform quick frequency counting. GenMax add pruning with mining and give the exact MFI (Maximal frequent itemsets) in two steps: (1st)-database is projected on current node, just like transaction and the mined MFI can also be projected on the node so fast superset checking. (2nd)- To perform fast support counting, GenMax uses Diffset propagation. Mafia [24] presented parent equivalence pruning (PEP) which is more effective pruning methods. To reduce the search space, Mafia also uses dynamic re-ordering. Both DepthProject and Mafia discover a superset of the MFI, and eliminate non-maximal itemsets by using post-pruning. MaxMiner and MafiaPP which is an extended version of Mafia are efficient in some datasets like *mushroom* dataset rather than GenMax. Smartminer [25] uses a heuristic function which uses the tail information (gathers and passes by Smartminer) to select the next node. It generates a smaller search tree needed a smaller number of supports counting and superset checking not needed. A new algorithm called as data stream mining for maximal frequent itemsets (DSM-MFI), which mine the set of all maximal frequent itemsets in windows over data streams was presented in [26]. An efficient algorithm for mining maximal frequent itemsets based on frequent pattern list named as (FPLMFI-Mining) [27]. FPLMFI-mining utilizes bit string and-operation to check maximal frequent itemsets. An algorithm based on a frequent pattern graph which used breadth-first search and depth-first-search techniques are used to generate all maximal frequent itemsets from the database [28]. A novel approach for finding the maximal frequent itemset from large data sources using the concept of segmentation of data source and prioritization of segments is proposed by M. Rajalakshmi, Dr. T. Purusothaman and Dr. R. Nedunchezhian [29]. Most of the association rule algorithms used to mine minimal frequent itemset first, then by using these minimal frequent itemsets mine the maximal frequent itemsets; these approaches are time consuming and large number of database scan required to discover the maximal frequent itemsets. To remove these problems, a new method to directly find the maximal frequent itemset by using the concepts of subsets is presented by Jnanamurthy HK, Vishesh HV, Vishruth Jain, Preetham Kumar and Radhika M. Pai [10]. The presented method adopted top-down searching strategy and it efficiently find maximal frequent itemsets. NVB Gangadhara Rao, Sirisha Aguru [30] proposed Hash Based Frequent Item sets-Double Hashing (HBFI-DH) in which vertical data format with hashing is used. Double hashing is used to avoid hash collision and secondary clustering problem. The advantages are fast access of data, easy to compute the hash function, efficiency and avoid unnecessary scans to the database. It avoids the primary clustering problem as well as secondary. G. Vijay Kumar and Dr. V. Valli Kumari [31] introduced a new single-pass algorithm called MaRFI (Maximal Regular Frequent Itemset) which mines maximal regular-frequent patterns in transactional databases using pair of transaction-ids instead of using item-ids. MaRFI mines the complete set of maximum regular frequent patterns at once in transactional databases using common items from transaction pairs that requires only one database scan. It is efficient than other algorithms that mine only for maximum frequent itemsets because it is very simple and easy to identify the common itemsets from transaction pairs and to calculate support and regularity threshold values. Maha Attia Hana [32] proposed a visualization of itemsets frequencies with matrix. The paper proposed a new method to extract maximal frequent itemsets called Matrix Visualization and Extraction of Maximal Frequent Itemsets .The constrained uncertain data maximal frequent itemset mining algorithm is proposed by Haizhou DU [33]. The proposed algorithm used to frequent itemsets mining, quantitative judgments that are further close to the objective and truthful thermal power unit running state can be made. Maximal frequent patterns are one of the condensed representations of frequent patterns. Recently, regular pattern mining along with frequent patterns playing an important role in data mining research. G. Vijay Kumar, V. Valli Kumari [34] presented a new algorithm called as IncMaRFI to mine MRF (Maximal regular frequent) itemsets in incremental databases in which new transaction is added continuously in old database, using common items from a set of transaction-id pairs. IncMaRFI algorithm extracts all the latest MRF itemset(s) at a time within a single scan.

V. DISCUSSION AND CONCLUSION

Association rule mining and Frequent pattern mining is currently very interesting and burning field for researchers due to theirs wild applicability. Association rule mining has a wide range of applicability such as cross marketing, market basket analysis, medical diagnosis and research, homeland security, Website navigation analysis, fraud detection and so on. Present paper provided the preliminaries of basic concepts about association rule mining and reviews the list of existing frequent pattern mining techniques. Some basic with improved and resent approach of maximal frequent itemset mining is also discussed. There are still many interesting research issues related to the modification and extensions of several approach like Apriori, FP growth, Eclat, MFI etc, such as structured patterns mining by further development of any of these approach, fault-tolerant patterns in noisy environments or approximate mining, frequent-pattern-based classification and clustering, and so on. Of course, a single paper cannot completely review all the techniques and approaches, yet it‘s hoped that the theoretical concepts and references given would guide the researcher in that research directions that have not been explored yet. In conclusion, MFI remains a promising and important algorithm, which would be used extensively by the researchers from different fields around the world.

REFERENCES

[1] M.S.V.K. Pang-Ning Tan, *“Data mining, in Introduction to data mining”*, Pearson International Edition, 2006, pp.2-7.

[2] J. Han, M. Kamber, “*Data Mining: Concepts and Techniques 3rd edition*”, Morgan Kaufmann Publishers, 2013.

[3] R. Agrawal and R. Srikant, “Fast algorithms for mining association rules,” in Proc. of Int. Conf. Very Large Data Bases (VLDB’94), Santiago, Chile, pp: 487–499, Sept. 1994.

[4] Wei Yong-Qing, Yang Ren-hua, and Liu Pei-yu., “An improved Apriori algorithm for association rules of mining,” IT in Medicine & Education, ITIME '09. IEEE International Symposium on, vol.1, pp.942-946, 2009.

[5] C. Zhang and J. Raun, “A Modified Apriori Algorithm with its application in Instituting Cross-Selling strategies of the Retail Industry,” in Proc. of 2009 International Conference on Electronic Commerce and Business Intelligence, pp: 515-518, 2009.

[6] W. Yu, X. Wang et al., “The Research of Improved Apriori Algorithm for Mining Association Rules,” in Proc. of 11th IEEE International Conference on Communication Technology Proceedings, pp: 513-516.

[7] D. Sun et al., “An algorithm to improve the effectiveness of Apriori Algorithm,” in Proc. of 6th ICE Int. Conf. on Cognitive Informatics, pp: 385-390, 2007.

[8] S. Bai and X. Dai, “An efficiency Apriori algorithm: P_matrix algorithm,” First International Symposium on Data, Privacy and Ecommerce, pp: 101-103, 2007.

[9] Z. Liu, G. Sang, and M. Lu, “A Vector Operation Based Fast Association Rules Mining Algorithm,” in Proc. of Int. Joint Conf. On Bioinformatics, System Biology and Intelligent Computing, pp: 561-564, 2009.

[10] Jnanamurthy HK, Vishesh HV, Vishruth Jain, Preetham Kumar, Radhika M. Pai, “Discovery of Maximal Frequent Item Sets using Subset Creation,” International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol. 3, No. 1, Jan. 2013.

[11] Don-Lin Yang, Ching-Ting Pan and Yeh-Ching Chung, “An efficient hash-based method for discovering the maximal frequent set,” Computer Software and Applications Conference, 2001. COMPSAC 2001. 25th Annual International, pp: 511-516, 2001.

[12] Han, J. Pei, J. , Yin,Y. and Mao,R., “Mining Frequent Patterns without Candidate Generation : A Frequent Pattern Approach,” in IEEE Transactions on Data Mining and Knowledge Discovery, Vol. 8, No. 1, pp: 53-87,2004.

[13] Grahne and J. Zhu, “Fast Algorithms for Frequent Itemset Mining Using FP-Trees”, IEEE Trans. on Knowledge and Data Engineering, vol. 17, no. 10, pp: 1347-1362, Oct. 2005.

[14] Savasere E. Omiecinski and Navathe S., “An efficient algorithm for mining association rules in large databases,” In Proc. Int’l Conf. Very Large Data Bases (VLDB), pp: 432– 443, 1995.

[15] Park. J.S, Chen M.S., Yu P.S., “An effective hash-based algorithm for mining association rules,” In Proc. ACMSIGMOD Int’l Conf. Management of Data (SIGMOD), pp: 175–186, 1995.

[16] C Toivonen. H., “Sampling large databases for association rules,” In Proc. Int’l Conf. Very Large Data Bases (VLDB), pp: 134–145, 1996.

[17] M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li., “New Algorithms for Fast Discovery of Association Rules,” Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD’97), AAAI Press, Menlo Park, CA, USA, pp: 283–296, 1997.

[18] C.Borgelt.” Efficient Implementations of Apriori and Eclat,” Proc. 1st IEEE ICDM Workshop on Frequent Item Set Mining Implementations (FIMI 2003, Melbourne, FL). CEUR Workshop Proceedings 90, Aachen, Germany 2003.

[19] Lin, D. and Kedem, Z.M., “Pincer-Search: An Efficient Algorithm for Discovering the Maximum Frequent Set,” in IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 3, pp: 553 – 566, 2002.

[20] R.J. Bayardo, “Efficiently mining long patterns from databases,” In SIGMOD, 1998.

[21] R. Agrawal, C. Aggarwal, and V. Prasad, “Depth first generation of long patterns, “In SIGKDD, 2000.

[22] K. Gouda and M. J. Zaki., “Efficiently mining maximal frequent itemsets,” In 1st IEEE Int'l Conf. on Data Mining, Nov. 2001

[23] Gouda, K., & Zaki, M. J, “GenMax: An Efficient Algorithm for Mining Maximal Frequent Itemsets,” Data Mining and Knowledge Discovery, Springer Science, 11, pp: 1-20, 2005.

[24] Burdick, D., Calimlim, M., and Gehrke, J. “MAFIA: A maximal frequent itemset algorithm for transactional databases,” In IEEE Intl. Conf. on Data Engineering, pp. 443–452, 2001.

[25] Zhou, Q. H., Wesley, C., & Lu, B. J. “Smart Miner: A depth 1st algorithm guided by tail information for mining maximal frequent itemsets,” In Proceedings of IEEE international conference on data mining, pp: 570–577, 2002.

[26] Hua-Fu Li, Suh -Yin Lee and Man-Kwan Shan, "Online mining (recently) maximal frequent itemsets over data streams," Research Issues in Data Engineering: Stream Data Mining and Applications, 2005. RIDE-SDMA 2005. 15th International Workshop on , pp: 11-18, 3-4 April 2005

[27] Jin Qian and Feiyue Ye, "Mining maximal frequent itemsets with frequent pattern list," Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on, vol.1, no., pp: 628-632, 24-27 Aug. 2007.

[28] Bo Liu and Jiuhui Pan, “A graph based algorithm for mining maximal frequent itemsets,” Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on, vol. 3, no., pp: 263-267, 24-27 Aug. 2007.

[29] M.Rajalakshmi,Dr.T.Purusothaman, Dr.R.Nedunchezhian, “Maximal Frequent Itemset Generation Using Segmentation Approach**”,** International Journal of Database Management Systems (IJDMS), Vol. 3, No.3, Aug 2011.

[30] NVB Gangadhara Rao, Sirisha Aguru, “A Hash based Mining Algorithm for Maximal Frequent Item Sets using Double Hashing,” Journal of Advances in Computational Research: An International Journal Vol. 1 No. 1-2 (Jan-Dec, 2012).

[31] G. Vijay Kumar, Dr. V. Valli Kumari, “MaRFI: Maximal Regular Frequent Itemset Mining using a pair of Transaction-ids”,International Journal of Computer Science & Engineering Technology (IJCSET), ISSN: 2229-3345, Vol. 4, No. 07, Jul 2013.

[32] Maha Attia Hana, “MVEMFI: Visualizing and Extracting Maximal Frequent Itemsets,” Int. Journal of Engineering Research and Applications Vol. 3, Issue 5, pp.183-189, Sep-Oct 2013.

[33] Haizhou DU, “An Algorithm for Mining Constrained Maximal Frequent Itemset in Uncertain Data,” Journal of Information & Computational Science Vol. 9, No. 15, pp: 4509–4515, 2012.

[34] G. Vijay Kumar, V. Valli Kumari, “IncMaRFI: Mining Maximal Regular Frequent Itemsets In Incremental Databases,” International Journal of Engineering Science and Technology (IJEST), ISSN: 0975-5462 Vol. 5, No.08, Aug. 2013.

[35] Agarwal, R., Imielinski T., and Swami, A. N., “Mining association rules between sets of items in large databases,” In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp: 207-216, 1993.

[36] Agarwal, R. Aggarwal, C. and Prasad V., “A tree projection algorithm for generation of frequent itemsets,” In J. Parallel and Distributed Computing, 2000.

[37] Wang, C., Tjortjis, C., “PRICES: An Efficient Algorithm for Mining Association Rules,” Lecture Notes in Computer Science, Vol. 3177,

pp: 352 – 358, Jan 2004.

[38] Yuan, Y., Huang, T., "A Matrix Algorithm for Mining Association Rules," Lecture Notes in Computer Science, Vol. 3644, pp: 370 - 379, Sep 2005.

[39] C Toivonen. H., "Sampling large databases for association rules," In Proc. Int'l Conf. Very Large Data Bases (VLDB), pp: 134–145, 1996.

[40] Dharmender Kumar, Naveen "Performance Analysis of Data Mining Algorithms to Generate Frequent Itemset," International Journal of Artificial Intelligence and Knowledge Discovery Vol.1, Issue 2, April 2011.

[41] Chuang, K., Chen, M., Yang, W., "Progressive Sampling for Association Rules Based on Sampling Error Estimation," Lecture Notes in Computer Science, Vol. 3518, pp: 505 - 515, Jun 2005.

[42] Li, Y., Gopalan, R., "Effective Sampling for Mining Association Rules," Lecture Notes in Computer Science, Vol. 3339, pp: 391 - 401, Jan 2004.

[43] Zaki, M. J., "Parallel and distributed association mining: A survey," IEEE Concurrency, Special Issue on Parallel Mechanisms for Data Mining, Vol. 7, no. 4, pp: 14 - 25, Dec. 1999.

[44] Cheung, D., Han, J., Ng, V., Fu, A. and Fu, Y. "A fast distributed algorithm for mining association rules," in `Proc. of 1996 Int'l. Conf. on Parallel and Distributed Information Systems', Miami Beach, Florida, pp: 31 - 44. 1996.

[45] Schuster, A. and Wolff, R., "Communication-efficient distributed mining of association rules," in `Proc. of the 2001 ACM SIGMOD Int'l. Conference on Management of Data', Santa Barbara, California, pp. 473-484. 2001.

[46] Cheung, D., Xiao, Y., "Effect of data skewness in parallel mining of association rules," Lecture Notes in Computer Science, Vol. 1394, pp: 48 - 60, Aug 1998.

[47] Manning, A., Keane, J., Data Allocation Algorithm for Parallel Association Rule Discovery, Lecture Notes in Computer Science, Vol. 2035, pp: 413-420.

[48] Parthasarathy, S., Zaki, M. J., Ogihara, M., "Parallel data mining for association rules on shared-memory systems," Knowledge and Information Systems: An International Journal, Vol. 3, no. 1, pp: 1–29, Feb. 2001.

[49] Tang, P., Turkia, M., "Parallelizing frequent itemset mining with FP-trees," Technical Report titus.compsci.ualr.edu/~ptang/papers/par-fi.pdf, Department of Computer Science, University of Arkansas at Little Rock, 2005.

[50] Wojciechowski, M., Zakrzewicz, M., "Dataset Filtering Techniques in Constraint-Based Frequent Pattern Mining," Lecture Notes in Computer Science, Vol. 2447, pp: 77 - 83, 2002.

[51] Tien Dung Do, Siu Cheung Hui, Alvis Fong, "Mining Frequent Itemsets with Category- Based Constraints," Lecture Notes in Computer Science, Vol. 2843, pp: 76 - 86, 2003.

[52] Das, A., Ng, W.-K., and Woon, Y.-K. "Rapid association rule mining," In Proceedings of the tenth international conference on Information and knowledge management. ACM Press, pp: 474-481. 2001

[53] QIAN Guangchao, JIA Ruiyu, ZHANG Ran, LI Longshu. "One Optimized Method of Apriori Algorithm," Computer Engineering, Vol. 34, no. 23, pp: 196-198. 2008;

[54] WANG Chengliang, WU Yanjuan, "Research and Application of Efficient Association Rule Discovery Algorithm of Chinese Medicine," Computer Engineering and Applications. Vol. 46, no. 34, pp: 119-122. 2010.

[55] ZENG Wandan, ZHOU Xubo, DAI Bo, CHANG Guiran, LI Chunping, "An Association Mining Algorithm Based on Matrix," Computer Engineering, Vol. 32, no. 2, pp: 45-47. 2006;

[56] ZHANG Yueqin, "Research of Frequent Itemsets Mining Algorithm Based on 0-1 Matrix," Computer Engineering and Design, vol. 30, no. 20, pp: 4662-4664. 2009;

[57] LV Taoxia, LIU Peiyu, "Algorithm for Generating Strong Association Rules Based on Matrix," Application Research of Computers, vol. 28, No. 4, pp: 1301-1303, 2011;

[58] ZHANG Yuntao, YU Zhilou, ZHANG Huaxiang, "Research on High Efficiency Mining Frequent Itemsets on Association Rules," Computer Engineering and Applications, Vol. 47, No. 3, pp: 139-141, 2011;

[59] Zhiyong Wang, "An Efficient Association Rules Algorithm Based on Compressed Matrix," TELKOMNIKA, Vol. 11, No. 10, pp: 5711 - 5717, Oct 2013.

[60] Harpreet Singh and Renu Dhir, "A New Efficient Matrix Based Frequent Itemset Mining Algorithm with Tags," International Journal of Future Computer and Communication, Vol. 2, No. 4, Aug. 2013.

[61] Pei. J, Han. J, Lu. H, Nishio. S. Tang. S. and Yang. D., "H-mine: Hyper-structure mining of frequent patterns in large databases," In Proc. Int'l Conf. Data Mining, 2001.

[62] Bin Fu, Eugene Fink and Jaime G. Carbonell, "Analysis of Uncertain Data: Tools for Representation and Processing," IEEE 2008.

[63] The Gartner Group, www.gartner.com.

TABLE I
SUMMARY OF FREQUENT PATTERN MINING ALGORITHMS

Algo. **Parameters**	**Apriori Algorithm**	**FP-Growth Algorithm**	**Sampling Algorithm**	**DHP Algorithm**	**Partitioning Algorithm**	**Eclat Algorithm**	**H-Mine Algorithm**
Data Structure	Array	Tree	Array	Array	Array	Array	Tree
Description	Use Apriori property, join and prune method	Construct conditional frequent pattern tree and conditional pattern base from database which satisfy the min. support	take any random sample and count support and validate with whole database at lower threshold support	Use hashing approach for fining frequent itemsets	find local frequent item first by Partition the database	Use set intersection. of transaction ids list for generating candidate itemsets	Partition and project the database; uses hyperlink pointers to store this database into main memory
Advantage	Basic algo. In mining frequent pattern, Suitable for both sparse and dense database	Only 2 scan of database, Suitable for large and medium datasets	Memory utilization and less time required; Suitable for any kind of dataset	Better than Apriori in small and medium database; Suitable for medium databases	Reduce the number of database scans, Suitable for large databases	Suitable for medium and dense datasets, time is small then Apriori algorithm	Better memory utilization, Suitable for sparse and dense datasets
Disadvantage	Large no. Of database scan, space and time complexity is high	recursive construction of the FP-trees and complex data structure required large	mostly not give accurate results	Not good for large database	more time is required because first find local frequent and then global frequent	not suitable for small datasets	time is larger than others because of partitioning of the database
Reference	[3], [11],[40]	[12], [13],[40]	[16]	[15]	[14]	[17], [18],[40]	[61]

TABLE II
SUMMARY OF VARIOUS FREQUENT PATTERN MINING ALGORITHMS

Algo. **Parameters**	**MBAT Algorithm**	**MaxMiner Algorithm**	**GenMax Algorithm**	**Smart - Miner Algorithm**	**The pincer – search Algorithm**	**MFIF Algorithm**	**MAFIA Algorithm**
Description	Find frequent itemset using Matrix with tag.	uses a breadth first search and a look ahead pruning strategy to find MFI (Maximal frequent itemset)	Use backtracking search approach, progressive focusing technique, and Diffset propagation.	Gathers and passes tail information and uses a heuristic function.	uses horizontal data format, Both top down and bottom up approach.	Use 2 dimension array, and subset approach to find Maximal frequent itemset First.	uses vertical data format, and Compression and projection of table.
Advantage	Only one scans of database and fast than Apriori Algorithm.	MaxMiner is the best for mining (T10 and T40) type database, additional pruning power.	Good for mining the exact set of maximal patterns, easily integrated with other algo.	Smartminer does not require any superset checking.	Very efficient when the longest frequent itemset of a database is big.	Single database scan, better memory utilization and.	Mine a superset of all maximal frequent itemsets.
Disadvantage	Large Database lead to large matrix required large space and execution time.	No. Of scan of database is same as Apriori algorithm, so large execution time.	Not work on large transaction database like 5M, 10M transactions.	Not work on large transaction database.	the initialization of the maximal Frequent candidate set is not efficient.	Only suitable where frequent itemset present at initial stage.	If the set of MFI is large, the Superset checking can be very expensive.
Reference	[7,8,9,60]	[20,62]	[22, 23]	[25]	[19]	[10]	[24]

Sukhbir Singh, Pursuing M.Tech (CSE) from Guru Jambheshwar University of Science & Technology, Hisar, Haryana, India; and received the master degree in Computer Application from GJUS&T Hisar, Haryana in 2012. His area of interest includes data mining, database system and computer networking.

Dharmender Kumar, is currently working as Associate Professor in Department of Computer Science & Engineering, Guru Jambheshwar University of Science & Technology, Hisar, Haryana, India. He did his Ph.D. from Guru Jambheshwar University of Science & Technology, Hisar, Haryana, M.Tech. (CSE) from Kurukshetra University, Kurukshetra and B.E. from Chhotu Ram State college of Engineering, Murthal (Sonipat). His area of interest includes database system, data mining and computational intelligence.