Latest Survey on Frequent Pattern Mining

M.Sinthuja[1], Dr. N. Puviarasan[2], Dr. P.Aruna[3]
[1, 3] Dept of Computer Science and Engineering
[2]Department of Computer and Information Science,
[1, 2] Annamalai University, Tamil Nadu

Abstract- *Frequent patterns mining is one of the most important concepts in data mining. In last decades, lots of research has been done in area of frequent pattern mining. Frequent patterns are used in many data mining task such as association rules, correlations, clusters etc. This paper surveys latest frequent pattern mining algorithms and compare them to know their disadvantages and advantages over others and to understand various problems still to be solved.*

Keywords- Apriori algorithm, Association rule mining, Data Mining, Frequent itemset mining, FP-growth algorithm

I. INTRODUCTION

Data mining is useful for gathering information from huge sets of data. In modern era, data mining has huge potential in IT industry and society due to wide availability of huge amounts of data. Data mining is the process of extraction of hidden information from the database and it can be viewed as steps in knowledge discovery process [1]. Data mining techniques finds its applicability in medicine, banking, retail sector, telecom etc. The main bottleneck in data mining is creating agile and efficient algorithms that can handle large volumes of data.

Nowadays, database practitioners and researchers pay their attention in data mining because of its uses in many areas such as decision support, market strategy and financial forecasts. Most of the people use the term “knowledge discovery device” or KDD for data mining. Data mining or knowledge discovery device includes,

1) Data cleaning: It removes noise and irrelevant data from the database.
2) Data integration: It combines multiple data sources into single data store.
3) Data Selection: Task relevant data are retrieved from data base.
4) Data transformation: Data is consolidated and transformed into standard formats appropriate for mining.
5) Data Mining: At this step, various data mining techniques and tools are applied in order to extract data pattern or rules.
6) Pattern evaluation: It involves thorough investigation of the knowledge finding tree pattern
7) Knowledge representation: The discovered knowledge is visualized using suitable method.

The knowledge extraction techniques available are association rule, classification, clustering, prediction and evaluation pattern etc. Among these techniques one of the most important approaches is mining association rules [2].

Association rule mining is one of the important among many techniques of data mining [3] [4]. It aims to extract frequent patterns, relations among a large set of data items. As a part of association rule mining, frequent patterns mining is that it extracts specific patterns with supports higher than or equal to a minimum support threshold from huge datasets. Association rules are commonly used in telecom industry, market analysis, stock management etc. Consider the supermarket where the owner may not know how the product is performing in the market and which products are sold frequently. So that they can determine the profit and loss of each product. In such cases mining method called frequent pattern mining algorithms called Apriori and FP-tree algorithms can be used to generate the frequently sold items. This is called as frequent pattern mining. So that the owner can enhance the user shopping model and also perform better marketing strategy of the products. Many methods have been proposed to reduce the number of association rules, pruning the search to rules which are found to be “interesting” non-repeating, or satisfying criteria such as leverage, coverage, strength or lift.

Two Step Approach of Association rule

i. Frequent Itemset Generation
Generate all item sets which satisfy minimum support
ii. Rule Generation

II. LITERATURE SURVEY

A. AIS Algorithm

The first algorithm introduced for frequent pattern mining is AIS algorithm. It is multi-pass algorithm in which candidate itemsets are generated during scanning the database by extending known-frequent itemsets with items from each transaction. There are mainly two disadvantages of the AIS algorithm. 1) It generates large number of candidates that later turn out to be infrequent. 2) The data structures were not specified.

B. SETM Algorithm

The SETM algorithm uses SQL to explore frequent itemsets. The SETM algorithm produces each member of the candidate and frequent itemset in the form of <TID, itemset> where TID is the different identification of a transaction. The limitation of SETM algorithm is that has to undergo multiple passes over the database. The major issue in SETM is the number of candidate itemsets. The SETM algorithm takes more space as TID is associated with each candidate itemset.

C. Apriori Algorithm

In the study of association rule mining Apriori is a classical algorithm. The performance of Apriori algorithm is better than AIS and SETM algorithm. Apriori entirely includes the subset count based pruning it means: it does not process any itemset whose subset is known to be frequent. It uses hash tree to store the frequency of the candidate itemset. The main drawback of Apriori algorithm is the generation of candidate itemset which consumes more time and memory. Another limitation is multiple scan of the database.

Improvements of Apriori is as follows: partitioning techniques [5], sampling approach [6], dynamic itemset counting [7], novel method for finding the presence of itemset [8], CARM [9], hashing technique [10], incremental mining [11], Efficient parallel mining for association rules [12], Mining sequential patterns [13], Parallel algorithm for discovery of association rules [14], an integrating mining with relational database systems [15] A tight upper bound on number of candidate patterns[16], Intersection Algorithm [17], Proposed Algorithm based on Apriori [17], DFPMT Algorithm [18], Improved Frequent Pattern Algorithm [19], DHP Algorithm [20]. Some of the improvements of Apriori algorithm are explained as follows.

D. Sampling Algorithm

In Sampling algorithm random samples of database is mined to find itemsets that are frequent within the sample. These itemsets are considered as a representative of the actual frequent itemsets in application where approximate mining results are sufficient. To obtain the exact mining result, this approach needs one or two scans over the entire database.As it use tuple by tuple approach it suffers from the drawback of Apriori algorithm.

E. DIC Algorithm

Another name of DIC algorithm is non-level-wise algorithm. In this algorithm, after each M transaction candidate itemsets are removed; here M is a parameter to the algorithm. It requires multiple passes over the transaction. The DIC algorithm is also suffers from the drawbacks of tuple-by-tuple approaches.

F. CARMA Algorithm

In CARMA (Continuous Association Rule Mining Algorithm) algorithm, the computation of frequent itemsets is online. It shows the present association rule to the user and allows the users to modify the parameters and also allows to change minimum confidence and minimum support at any transaction during the initial scan of the database. It requires two passes over the algorithm. This algorithm explores and remove candidate after each record of the database is processed. The performance of the algorithm is not constantly better than Apriori algorithm as its consumption of memory is lower by an order of magnitude.

Intersection Algorithm

In Intersection approach, by using the intersect query of SQL the count is calculated from common transaction that contains in each elements of candidate set. This approach consumes lower amount of time when compared to classical Apriori algorithm.

G. Proposed Algorithm based on Apriori

This algorithm adopts the concepts of Record filter approach and Intersection approach in Apriori algorithm. By using the intersect query of SQL support calculated by searching for common transaction that present in every elements of candidate set. Restriction is applied in considering the transaction. Proposed algorithm proved that it is better than other frequent pattern mining algorithms of Apriori and Intersection algorithm.

H. DFPMT Algorithm

Dynamic Approach for Frequent Patterns Mining using Transposition (DFPMT) of database is for mining frequent patterns which are based on Apriori algorithm. It uses

Dynamic function for Longest common Subsequence. In DFPMT, the database is stored is transposed form and for each iteration database is filtered by exploring LCS of transaction id for each pattern.

I. Improved Frequent Pattern Algorithm

By modifying Apriori-like algorithm the improved frequent pattern algorithm mines frequent pattern from large dataset using transposition of the database. The major advantage of the algorithm is: database is stores in transposed form and for each iteration database is filtered and reduced by exploring the transaction id for each pattern. Thus the algorithm reduces the huge consumption of time and reduces the database size.

J. DHP Algorithm

Direct Hashing and Pruning (DHP) method proposes two important optimization techniques to speed up the algorithm. In the first optimization technique candidate itemsets are pruned for each iteration. Second optimization is to trim the transactions to make the support-counting process more efficient.

K. FP-Tree Based Algorithm

FP-growth algorithm mines the entire set of frequent itemsets without candidate itemset generation. It adopts divide and conquer methods. The transaction database is compressed into a frequent pattern tree called as FP-tree. Frequent patterns can be mined from FP-tree. The drawback of FP-tree is tree construction which consumes more time. The second drawback is not flexibility and reusability during the process of mining. There are many improvements to FP-growth approach, including depth-first generation of frequent itemsets, an array based implementation of prefix tree structure for efficient pattern growth mining, CATS algorithm, AFPIM algorithm, CAN tree, CP-tree, Efficient Prefix tree. Following are some improvement of FP-growth algorithm in brief.

L. CATS Algorithm

In an incremental manner CATS algorithm mines frequent patterns. CATS algorithm is an improvement of FP-tree. It explores frequent patterns without candidate itemset generation. In this algorithm, the first transaction in database is added to the tree's root. For subsequent transactions, the items within the transaction are compared with the items in the tree to identify shared items. If there is any item in common between tree nodes and the transaction, the transaction is merged with the node that has the highest frequency level. Then, the remainder of the transaction is added to the merged nodes. This process is recursively repeated until all common items are discovered.

M. AFPIM Algorithm

In AFPIM algorithm, PreMinsup is considered whose values are set less than the Minsup. Since, items are ordered based on the number of events, the insertion, deletion or modification of transactions may affect the frequency and order of the items. Items in the tree are adjusted when the order of the items changes. The AFPIM algorithm swaps such items by applying bubble sort method that involves huge calculation [28].

N. CanTree Algorithm

Only one database scan is required in Can-tree. In this algorithm items are ordered in alphabetical order which can be determined by the user. If there is any changes in frequency like insert, delete or update, it will not affect the order of items in Can-tree. Thus, new transaction can be inserted into Can-tree without swapping any tree node.

O. CP-Tree Algorithm

Based on predefined order all the transactions are inserted into the Can-tree algorithm. In this algorithm, a list is created to maintain a predefined item order, called I-list. After inserting some of the transactions, if the item order of the I-list differs from the current frequency-descending item order to a predefined degree, the CP-tree is restructured by method called the branch sorting. Then, the item order is updated with the current list [30].

Efficient Prefix Tree

The problem with CP-tree is at the time of construction of CP-tree items are ordered in descending order of previous insertion phase, thus its restructuring is very costly. To overcome this problem, efficient prefix tree structure is used to decrease the time of restructuring.

P. MAXCLIQUE Algorithm

The above discussed algorithms mine the database which is in horizontal data layout. The other way of mining, in which data presented in vertical data format (i.e., {item: TID_set}). Each item is shown with list of TIDs (Transaction Ids), in which item appears. The MaxClique algorithm is designed to efficiently mine databases which are in a vertical layout [14].

Q. Equivalence Class Transformation (ECLAT)

The ECLAT algorithm uses vertical (or inverted) layout which consists of a list of items, with each item followed by its Tid-list the list of all the transaction identifiers containing the item. ECLAT algorithm accommodates ‘Depth First Search’ approach and requires the generation of candidate itemset. The ECLAT algorithm was constructed to control the shortcomings of the count and candidate distribution algorithms. It uses the aggregate memory of the system by partitioning the candidates into disjoint sets using the equivalence class partitioning. Drawback of this algorithm is huge number of candidate itemset generation.

R. Viper

Previous vertical mining algorithms have many restrictions regarding database shape, size, and mining process. Viper algorithm does not undergo any restriction. Many optimizationis included to enable efficient processing. The viper algorithm outperforms other vertical mining algorithms.

S. Pattern Mining Algorithm (PM)

Pattern mining algorithm adopts simple processing technique to generate frequent patterns. Only one database scan is required in this algorithm. The two phase of pattern mining algorithm are: i) Explores subset list ii) Explores frequent patterns using subset list. Drawback is consumption of runtime in generating frequent patterns.

III. CONCLUSION

Previously, many researchers have developed various algorithms, compared them and attempt to resolve the frequent itemset problem effectively in terms of minimum number of database scans, consumption of time and memory. In this survey paper, a detailed discussion about the existing algorithms of frequent pattern mining in both data layout such as horizontal and vertical is attempted. Each algorithm is having its own advantages and disadvantages. Some of the algorithms work on vertical layout and some on horizontal layout. This survey paper has also highlighted many algorithms which made a significant contribution to improve the efficiency of frequent pattern mining.

REFERENCES

[1] J. Shrivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web Usage mining: Discovery and applications of Usage patterns from web data. SIGKDD Explorations, 1(2), 2000.

[2] R. Agrawal, T. Imielienski, and A. Swami. 1993. Mining Association Rules between Sets of Items in Large Databases. Proc. Conf. on Management of Data, 207–216. ACM Press, New York, NY, USA 1993.

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

[4] R. Agrawal and R. Srikant. 1994. Fast Algorithms for Mining Association Rules. Proc. 20th Int. Conf. on very Large Databases (VLDB 1994, Santiago de Chile), 487–499. Morgan Kaufmann, San Mateo, CA, USA 1994.

[5] A. Savasere, E. Omieccinski and S. Navathe, 1995. An efficient algorithm for mining association rules in large databases. Proceedings of the 21st International Conference on Very Large Databases, September 11-15, 1995, Zurich, Switzerland, pp: 432-443.

[6] H. Toivonen, 1996. Sampling large databases for association rules. Proceedings of 22th International Conference on Very Large Databases, September 3-6, 1996, Bombay, India, pp: 134-145.

[7] S. Brin, R. Motwani, J.D. Ullman and S. Tsur, 1997. Dynamic itemset counting and implication rules for market basket data. Proc. 1997 ACM SIGMOID Int. Conf. Manage. Data, 26: 255-264.

[8] A. Tiwari, R.K. Gupta and D.P. Agrawal, 2009. A novel algorithm for mining frequent itemsets from large database. Int. J. Inform. Technol. Knowl. Manage., 2: 223-229.

[9] C. Hidber, 1999. Online association rule mining. ACM SIGMOD Rec., 28: 145-156.

[10] J.S. Park, M.S. Chen and P.S. Yu, 1995. An effective hash-based algorithm for mining association rules. ACM SIGMOD Rec., 24: 175-186.

[11] D.W. Cheung, J. Han, V.T. Ng and C.Y. Wong, 1996. Maintenance of discovered association rules in large databases: An incremental updating technique. Proceedings of International Conference on Data Engineering, February 26-March 1, 1996, New Orleans, Louisiana, pp: 106-114.

[12] J.S. Park, M.S. Chen and P.S. Yu, 1995. Efficient parallel mining for association rules. Proceedings of the 4th International Conference on Information and Knowledge Management, Nov. 29-Dec. 2, Baltimore, MD., pp: 31-36.

[13] R. Agrawal and R. Srikant, 1995. Mining sequential patterns. Proceedings of the 11th International Conference

on Data Engineering, March 6-10, 1995, Taipei, Taiwan, pp: 3- 14.

[14] M.J. Zaki, S. Parthasarathy, M. Ogihara and W. Li, 1997. Parallel algorithm for discovery of association rules. Data Min. Knowl. Discov. 1: 343-374.

[15] S. Sarawagi, S. Thomas and R. Agrawal, 1998. Integrating association rule mining with relational database systems: Alternatives and implications. ACM SIGMOD Rec.,27:343- 354.

[16] F. Geerts, B. Goethals and J. Bussche, 2001.A tight upper bound on the number of candidate patterns. Proceedings of the 2001 International Conference on Data Mining, Nov. 29-Dec. 2, San Jose, CA., pp: 155-162.

[17] D. N. Goswami, Anshu Chaturvedi, C. S. Raghuvanshi, 2010. An algorithm for frequent pattern mining based on Apriori. International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 942-947.

[18] S. Joshi. An Implementation of Frequent Pattern Mining Algorithm using Dynamic Function. International Journal of Computer Applications (0975 – 8887) Volume 9- No.9, November 2010.

[19] D. Gunaseelan, P. Uma. An Improved Frequent Pattern Algorithm for Mining Association Rules. International Journal of Information and Communication Technology Research, Volume 2 No. 5, May 2012, ISSN 2223-4985.

[20] J. S. Park, M. S. Chen, and P. S. Yu. An effective hash based algorithm for mining association rules. In Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, volume 24(2) of SIGMOD Record, pages 175–186. ACM Press, 1995.