International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-8 Issue-11, September 2019

A Proficient Algorithm For Mining Frequent Item Sets

M Vanitha, D Narasimhan, Mehul P Barot

Abstract*— *Frequent Item set Mining (FIM) discover recurrent item sets that are extremely associated in a transactional database. It can be used in big data applications like gene extraction, social network analysis, IOT devices sensor data analysis, etc. As the data size keeps on growing, an efficient procedure is necessary to process the enormous volume of data. Apriori is one of the topmost algorithm and its entry enhanced the research in mining. The algorithm requires multiple scans of the database and produce more candidate items, which increase the computation time and storage along with the transactions size. An efficient technique is required to boost the performance of the algorithm in terms of storage and computational complexity. To reduce the complexity we proposed a reduction factor to Apriori and it can be adopted before candidate generation. Experimental results revealed that our proposed technique greatly reduce the total execution time as well as storage requirements.

Keywords*— *Frequent Item set Mining, Apriori algorithm, Association rule

I. INTRODUCTION

Computer systems generate huge volume of data. To extract significant information, knowledge, pattern or statistics from the large volume of data certain methodologies are required. Manual mining of data happened for centuries. When the size of the data tends to become large it is very difficult for a human to discover the information accurately. A computer can be used to automate the process and it leads to a field of computer science called data mining [1]. Data mining carry out analysis on the data and employ software techniques to extract meaningful information. Extracted information can be used further for various purposes.

Pattern mining or FIM [2] is a dominant subfield of data mining. FIM consists of procedures to find out interesting and useful patterns from different databases. Interesting patterns are nothing but frequent patterns. A pattern which occurs more number of times in a transaction or a database is called frequent pattern. FIM plays a vital part in mining fascinating associations among data. Additionally, it aids in classification, data indexing, clustering, and other data mining tasks as well. FIM is an essential task and a motivated subject in data mining research. After finding interesting patterns, association between the patterns or items can be found using Association Rule Mining (ARM) [3]. It is a rule based machine learning algorithm expected to identify the rules. ARM estimate the existence of precise item based on the occurrences of other items in the database. FIM and ARM can be applied to various domains like Market Basket Analysis, Biological data analysis, Social Network Analysis,, Sentimental analysis, Click stream analysis, Clustering, Classification, Software bug detection, Recommendation systems, Prediction, Web mining, etc.

Several algorithms exists for Frequent Item set and Association rule mining. Among them well-known are Apriori, FP-growth and Éclat. Apriori [4] is the classic, familiar one and easy to implement. Rakesh Agrawal and Ramakrishnan Srikant introduced this algorithm in 1994. It is an iterative algorithm and produce the frequent item set in two steps: Frequent item set generation and Rule generation. Since the algorithm use earlier information of recurrent item set properties, it is named as A-priori. It isolate the repeated individual items in the file (singletons). From the singletons further combinations are made and its occurrence will be extracted until no additional combination exists.

FP growth algorithm is an enhancement of Apriori algorithm, introduced by Han [5] in 2000. It discovers the frequent item set from a transaction database without candidate generation hence increase performance. Divide and conquer strategy is adopted. The main essence of the process is that it use an extraordinary data structure called Frequent Pattern tree. The tree will compress the database and store the frequent items sets and its association. Like Apriori the algorithm for FP-growth consists of two steps: Construction of FP-Tree and Extract frequent item sets. It decrease search cost by observing short patterns recursively and then combine the short patterns into long frequent patterns, proposing good selectivity.

Compared to Apriori, FP-growth has some advantages. It use compressed data structure, eliminate candidate generation, use divide and conquer approach and scan the database only twice. Beneath the advantages, the algorithm has some drawbacks: FP-Tree may not fit in memory, construction of FP-Tree is expensive and takes more execution time since it follows a complex data structure.

Eclat[6] was suggested by Zaki. Contrast to Apriori & FP-growth, depth-first search is used in Eclat to find repeated item sets instead of breadth-first search. Eclat means Equivalence Class clustering and bottom up Lattice Traversal.

Manuscript published on 30 September 2019.
***Correspondence Author(s)**
M Vanitha, Department of CSE, SASTRA Deemed to be UniversityKumbakonam, Tamilnadu, India,
(Email: vanitha_mu@src.sastra.edu)
D Narasimhan, Department of Mathematics, SASTRA Deemed to be University, Kumbakonam, Tamilnadu, India,
(Email: narasimhan@src.sastra.edu)

This algorithm use vertical data format for a database instead of horizontal format as shown in figure 1. Each transaction has a unique id called transaction id denoted as TID. Entire database scanned only once and a tid list for each item will be created. Each item occurrence can be computed by the intersection of tid set. Like Apriori, Eclat follows candidate generation.

HORIZONTAL DATA FORMAT

TID	Items
1	A, B, E
2	B, D
3	C, E, F
4	A, C, D, G
5	A, B, C, D
6	A, E
7	A, B, G
8	A, B, C
9	A, C, D
10	B, F

VERTICAL DATA FORMAT

ITEM	TID LIST
A	1, 4, 5, 6, 7, 8, 9
B	1, 2, 5, 7, 8, 10
C	3, 4, 5, 8, 9
D	2, 4, 5, 9
E	1, 3, 6
F	3, 10
G	4, 7

Fig. 1 Representation of transaction database

Eclat algorithm executes faster than Apriori, since a single database scan is required. Memory requirements also get reduced by depth-first search. But the algorithm can be applied only to datasets which are small in size.

Even though various algorithm exists for FIM, all of them suffer from computation, time and storage complexity. Our proposed algorithm target to diminish the total computations and obviously reduce the execution time.

The rest of the paper is structured as follows. Section II describe the basic concepts and related work. Our proposed scheme is enlightened in Section III. Data sets used in investigation and the outcome obtained are discussed in Section IV. Conclusion is outlined in Section V.

II. BASIC CONCEPTS & RELATED WORK

This section describe the basic terminologies which are required for the proposed scheme and numerous research work done before related to our scheme.

A. Frequent Item Set

Items that are bought by the customer are included in transaction. A database comprises of more number of transactions. An item is said to be frequent if it is included in most of the transaction. The set of items which are present in more number of transaction is considered as a frequent item set. Let D = { T_1, T_2, T_3, … , T_n } be the transactions a database includes, where T_i indicates a transaction and let I = { i_1, i_2, i_3, … , i_m } as the set of items in a transaction.

B. Association rule

The rules [3] are used to identify the relationship between the items which belongs to frequent item set. They are represented as if-then statements. The rules are used to envisage the occurrence of an item, based on the occurrence of further items in database. It consists of two parts:

i. Antecedent (if - X)

ii. Consequent (then - Y)

The rule can be represented as follows:

X => Y, where X, Y $\subseteq$ I

Example rule for a supermarket can be

{Bread, Butter} → {Milk}

The rule illustrates that if a customer bought Bread and Butter there is a chance to buy Milk also.

If a transaction contains more number of items then plenty of association rules can be generated from them. In order to extract effective and interesting rules, certain measures can be used. They are Support, Confidence and Lift.

C. Support

Support represents the frequency of the item set that appears in the database.

$$Sup(X) = \frac{Number\ of\ transactions\ containing\ X}{Total\ number\ of\ transactions}$$

D. Confidence

Confidence represents the frequency of the rule in the database.

$$Conf(X => Y) = \frac{Sup(X\ U\ Y)}{Sup(X)}$$

E. Lift

The downside of the *confidence* is that occasionally it may misrepresent the relationship between the items. Since it account on how popular the antecedent is and not the consequent. In general if consequent is popular then there will be a greater chance that a transaction holding antecedent will also cover consequent, thus boost the confidence measure. To consider the popularity of both, a third measure called Lift also called interestingness measure can be used.

$$Lift(X => Y) = \frac{Sup(X\ U\ Y)}{Sup(X)\ x\ Sup(Y)}$$

F. Related work

Apriori is a sequential algorithm, find the frequent item set and generate large number of candidates. To reduce the intermediate candidates, storage space as well as to increase the access efficiency various data structures are preferred by various authors. Transactions are represented as a Boolean matrix [7], where each column of a matrix represents an item. Frequent item sets are generated by matrix multiplication.

Hash based [8] candidate generation use a predefined hash function and it significantly improve the item access thus reduce space and time. Trie [9] follow a rooted directed tree structure to store candidates, easy approach to maintain as well as a faster technique. Hash table trie [10], trie structure along with hashing technique speed up the search time. Our proposed scheme use hash table to store the transaction item sets.

Retrieval Number: K24920981119/19©BEIESP
DOI: 10.35940/ijitee.K2492.0981119
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

Database can be divided into blocks and mining done on the partitions [11], local frequent patterns are found then combined into a global one. This approach moderate the amount of database scans and efficiency of the algorithm can be enriched to some degree. But the intermediate results produced for each block will be high. Extension of FP-tree called Fuzzy FP-tree [12] proposed to extract frequent fuzzy patterns. Guaranteed results and less execution time can be drawn only for minimal support threshold.

III. PROPOSED SCHEME

One of the crucial step while mining association is the generation of item sets that are frequent in the transaction. Our proposed scheme adopts Apriori algorithm. Despite the various shortcoming still Apriori is popular and easy to implement. Assume a dataset consists of N frequent items or singletons then total number of candidate item set generated from singleton will be 2^N-1. It requires O((2^N-1) x N x M) comparisons, where M represents the number of transactions. Hence complexity of the algorithm increases if singletons tends to be very large.

To reduce further combinations and comparisons, Anti-monotone property also called as Apriori property can be used as a pruning strategy. It states that

i. All subsets of a frequent item set are frequent
ii. Any superset of an infrequent item set is also infrequent

For very large data sets, the specified pruning property alone is not enough. Our proposed scheme introduce a reduction factor which is used to reduce the candidate generation after finding one frequent item sets. Using Apriori algorithm produce one frequent item set as shown in figure 2. Initially items are filtered using the minimum support measure and for the given example it is considered as 30%. Since the size of 1-frequent item set is 5 and it will generate 2^5-1 = 31 candidates (2-frequent item set).

If data set size is huge it may produce more candidates which may not fit in memory, further it leads to more computation. To reduce the above said difficulty, a reduction factor can be applied to 1-frequent item set. For this example it is considered as 60%. Reduction value will be calculated for each item by computing the ratio of maximum occurrence count and its original occurrence count. If reduction value is greater than are equal to factor value then the item can be selected as a candidate, otherwise rejected. For instance Item A's reduction value = (7 / 7) = 1 which is greater than 0.6, so A is selected as a candidate. Item E's reduction value = (3 / 7) = 0.428 which is less than 0.6, so E is rejected. Further transaction reduction is performed by removing non-frequent items from the database as well as the transactions which consists only infrequent item sets are entirely removed from database.

IV. EXPERIMENTS & RESULTS

The proposed scheme has been implemented using Java. Experiments are carried out for both synthetic and real datasets. A computer with 6th Generation Intel core i7-600 Quad core 3.4 GHz processor, 16GB primary memory, running Ubuntu 16.4 operating system has been used. Two datasets namely Foodmart (real) and T10I4D100K (synthetic) were considered for examination. Foodmart was obtained from Microsoft foodmart SQL Server 2000. The various characteristics of dataset are illustrated in Table 1.

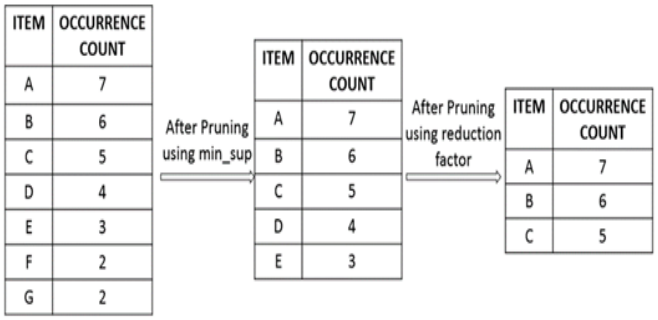

Fig 2. Our proposed scheme

Table 1: Dataset Characteristics

Sl. No.	Data Set	Total transactions	Total Distinct items	Average items per transaction
1	Foodmart	21,556	1559	4
2	T10I4D100K	1,00,000	870	10.1

The ultimate goal of our proposed work is to reduce the total execution time taken for rule generation by shrinking the candidates produced after singletons. Experiments were repeated for a set of minimum support value, several trials has been made for each minimum threshold value, average execution time is taken into consideration and some of them were plotted in graph. From the results we observed that the suggested scheme harvests good performance than the existing one. Figure 4a and 4b illustrates that our proposed scheme had taken less time when compared to traditional Apriori algorithm.

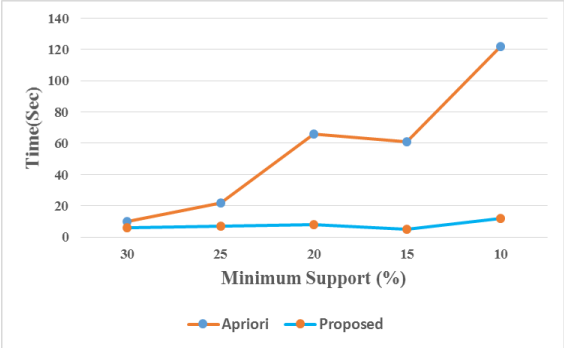

Fig. 4a Execution time for Foodmart dataset

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

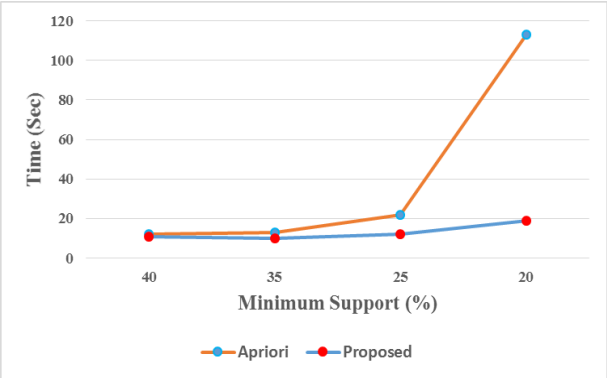

Fig. 4b Execution time for T10I4D100K dataset

IV. CONCLUSION

Frequent Item set Mining is an approach to find repeated patterns from a transaction database and Apriori is a popular mining algorithm. But it requires more computation as well as memory requirements. In order to reduce the drawbacks an efficient technique called reduction factor is presented. Transaction reduction is also accomplished to enhance the performance the algorithm. From the experiment results it is witnessed that our proposed scheme reduce the storage and time complexity of mining.

REFERENCES

1. Witten, I. H., Frank, E., and Hall, M. A. Data Mining - Practical Machine Learning Tools and Techniques. Morgan Kaufmann, USA, 2011.
2. M.M.J. Kabir, S.X. Xu, B.H. Kang, Z.Y. Zhao, Comparative analysis of genetic based approach and apriori algorithm for mining maximal frequent item sets, in: IEEE Congress on Evolutionary Computation, CEC, 2015 May 25–28, Sendai, Japan.
3. Agrawal, R., Imielinski, T. and Swami, A. 1993. Mining Association Rules between Sets of Items in Large Databases. In ACM SIGMOD Conf. Management of Data, Washington, D.C., 207–216
4. Agrawal, R. and Srikant, R. 1994. Fast Algorithms for Mining Association Rules. In Proceedings of the Twentieth International Conference on Very Large Databases, Santiago, Chile, 487–499
5. J. Han, H. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In: Proc. Conf. on the Management of Data (SIGMOD'00, Dallas, TX). ACM Press, New York, NY, USA 2000.
6. M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, ''Newalgorithms for fast discovery of association rules,'' in *Proc.KDD*, vol.97, 1997, pp. 283–286.
7. X. Luo and W. Wang, "Improved Algorithms Research for Association Rule Based on Matrix," 2010 International Conference on Intelligent Computing and Cognitive Informatics, pp. 415–419, Jun. 2010
8. Libing Wu, KuiGong, Fuliang Guo "Research on Improving Apriori Algorithm Based on Interested Table", IEEE, 2010
9. Bodon, F. and Rónyai, L. "Trie: an alternative data structure for data mining algorithms", Mathematical and Computer Modelling, 2003, 38(7), 739-751
10. Bodon, F. 2010. A fast apriori implementation. In Proceedings of IEEE ICDM workshop on frequent itemset mining implementations (FIMI'03), Vol. 90
11. Y.B. Jia, G.H. Xia, H.D. Fan, Q. Zhang, X. Li, An improved apriori algorithm based on association analysis, in :2012 Third International Conference on Networking and Distributed Computing (ICNDC), 2012 October 21–24, Hangzhou, China.
12. C.W. Lin, T.P. Hong, W.H. Lu, Linguistic data mining with fuzzy fp-trees, Expert Syst. Appl. 37 (2010).