IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 3, Ver. II (May – Jun. 2015), PP 105-109
www.iosrjournals.org

Analytical Study and Newer Approach towards Frequent Pattern Mining using Boolean Matrix

Paresh Tanna[1], Dr. Yogesh Ghodasara[2]
[1](PhD Scholar, School of Computer Science, RK University, India)
[2](Associate Professor, College of Information Tech., Anand Agriculture University, India)

Abstract : *The difficulty of association rule mining for objects in a massive database is to locate huge frequent patterns and relation among objects in a pattern from database entries has been examined with various different algorithms. But Apriori algorithm include plenty of challenges like vast amount of database check for creating big pattern and doing support calculation, great number of candidate findings. These comprised in this paper and impact to these a newer algorithm FPMBM (Frequent Pattern Mining with Boolean Matrix) has been considered. This newer algorithm uses a boolean matrix k-pattern for all patterns materialize in transactions. A list is maintained to short out the number of loops for patterns creation as well as only single database scan is followed in advance stage i.e. at the time of vertical database conversion.*
Keywords - *Association rule, Frequent pattern mining, Boolean Matrix*

I. Introduction

Association rules are provisional declarations that assists expose relationships between seemingly unrelated data in a row level DB [1]. A case of an association rule might be "If a buyer procures a Wafer, he is 60% expected to procure Chocolate" [2]. An association rule encompass 2 fractions, a predecessor (if) and a subsequent (then). A predecessor is an object originates in the data. A subsequent is an object that is originated in combination with the predecessor [1]. Researchers can focal point on frequent patterns mining like Frequent patterns from the little and/or from the huge quantity of facts, where the facts are either transactional or relational [7].

II. Worked Out Methods: Frequent Pattern Finding Techniques

Four major frequent pattern mining approaches are: Apriori[2], Direct Hashing and Pruning (DHP)[3], Frequent pattern growth (FP-Growth)[4], Vertical data format approach (ECLAT)[5] have been proposed to work with transactional database. Each one is having some contradicts with each other and finally justifies that a newer approach is there to outcome these contradicts.

III. Proposed System: FPMBM

In this paper we proposed a newer algorithm makes good use of Boolean matrix.

The FPMBM Algorithm (Frequent Pattern Mining with Boolean Matrix):

FPMBM Algorithm can be exploited for proficient large frequent pattern creation. FPMBM uses vertical data layout for creating huge frequent patterns from the database entries. We have find some contradicts for some frequent pattern mining algorithms like vast amount of database check for creating big pattern and doing support calculation, great number of candidate findings. With above analytics we can find some improvements like diminish passes of database scanning, shrivel quantity of candidates, assist support counting of candidates without database scanning[6][8]. By considering analytical study on these factors, we have experimented with Boolean matrix of vertical data layout rather than horizontal data layout. That is each row for pattern and each column for a particular entry in DB. In this way we put '1' for each item entry found in particular entry and put '0' for non entry. For this it becomes very much easier to find support count for k-pattern, perform summation of 1's only and no need to scan database each time for support counting. Also FPMBM maintains a list for "number of iterations" i.e. useful for K+1 itemsets generation. By using this list, limited and essential number of iteration will be followed by k-itemsets to generate k+1 –itemsets.

Algorithm : FPMBM

Steps:

1. Convert list of entries in D into Vertical Data Layout D_Ver
2. Generate Boolean Matrix from D_Ver. Maintain a list for no. of iterations i.e. useful for K+1 patterns creation.
3. Find Frequent 1-patterns from the given Boolean Matrix by summing 1 for each row i.e. for each pattern

4. Increment K by one
5. Perform candidate generation for K-patterns and also make Boolean Matrix for K-patterns. Maintain a list for no. of iterations i.e. useful for K+1 patterns generation.
6. Find large patterns by minSupport
7. Repeat through step-4 Until Boolean Matrix is null for all candidate creation

Example 1 :

We follow an example of a simple database with 9 entries and min_sup is 2 as shown below.

Table 1 : Entry Table

TID	List of itemIDs
ED1	X1,X2,X5
ED2	X2,X4
ED3	X2,X3
ED4	X1,X2,X4
ED5	X1,X3
ED6	X2,X3
ED7	X1,X3
ED8	X1,X2,X3,X5
ED9	X1,X2,X3

i) Vertical format from horizontal data
Convert horizontally layout entry into vertical layout by checking the database fist time.

ii) Boolean matrix, support count and 1-pattern, no.of iterations list creation
Create Boolean matrix for the above data and find frequent 1-patterns from the given Boolean Matrix is simply by summing 1 for each row i.e. for each pattern

Table 2 : Boolean Matrix with Vertical Layout of Table 1

Trans ID / Itemset	ED1	ED2	ED3	ED4	ED5	ED6	ED7	ED8	ED9	Support Count	No.of Iterations
X1	1	0	0	1	1	0	1	1	1	6	4
X2	1	1	1	1	0	1	0	1	1	7	3
X3	0	0	1	0	1	1	1	1	1	6	2
X4	0	1	0	1	0	0	0	0	0	2	1
X5	1	0	0	0	0	0	0	1	0	2	0

iii) AND operation

a. Create candidate set using AND operation and find support count for them for frequent 2-patterns from the given Boolean Matrix by ignoring sup count < min_sup.

Table 3 : AND Operations with itemsets of Table 2

Trans ID/ Itemset	ED1	ED2	ED3	ED4	ED5	ED6	ED7	ED8	ED9	Support Count	No. of Iterations
(X1,X2)	1	0	0	1	0	0	0	1	1	4	2
(X1,X3)	1	0	0	0	0	0	1	1	1	4	1
(X1,X5)	1	0	0	0	0	0	0	1	0	2	0
(X2,X3)	0	0	1	0	0	1	0	1	1	4	2
(X2,X4)	0	1	0	1	0	0	0	0	0	2	1
(X2,X5)	1	0	0	0	0	0	0	1	0	2	0

b. Create candidate set using AND operation and find support count for them for frequent 3-patterns from the given Boolean Matrix by ignoring sup count < min_sup.

Table 4 : AND Operations with itemsets of Table 3

Trans ID/ Itemset	ED1	ED2	ED3	ED4	ED5	ED6	ED7	ED8	ED9	Support Count	No. of Iterations
(X1,X2,X3)	0	0	0	0	0	0	0	1	1	2	1
(X1,X2,X5)	1	0	0	0	0	0	0	1	0	2	0

This method recurs, for k increased by 1 regularly, until no frequent patterns or no items for patterns creation could derive.FPMBM uses a temporary list for no. of iterations. Using this list, FPMBM controls no. of iterations for candidate creation. FPMBM uses very less no. of iterations compare to other frequent pattern mining algorithms discussed here.

IV. Experimental Results

We compared the concert of the newer technique with various different algorithms as shown in Fig 1 to 5. The newer technique is implemented in Java and compiled with java compiler with use of Netbeans IDE 6.8. We choose the dataset from [9] for testing the concert of the new technique. All datasets are direct or indirect which are taken from the FIMI repository page. Table 5 below illustrate characteristics of those data sets. Experiments were carried out on PC equipped with a Win-8, core i3 processor and RAM 2 GB. Execution times (in seconds) required by different algorithms are shown in below figure 1, 2, 3, 4 and 5.

Table 5. Characteristics Of Datasets Used For Experiment Evaluation

[1 – Apriori, 2 – DHP, 3 – ECLAT, 4 – FP Growth 5 – IFPMBM(New Algorithm)]

Dataset	Records	Algorithms Comparisons	Remarks
T10I4D100	100	1,2,3,4,5	Top 100 records from T10I4D100K
T10I4D1000	1000	1,2,3,4,5	Top 1000 records from T10I4D100K
T10I4D10000	10000	3,5	Top 10000 records from T10I4D100K
T10I4D50000	50000	3,5	Top 50000 records from T10I4D100K
T10I4D100K	100000	3,5	-

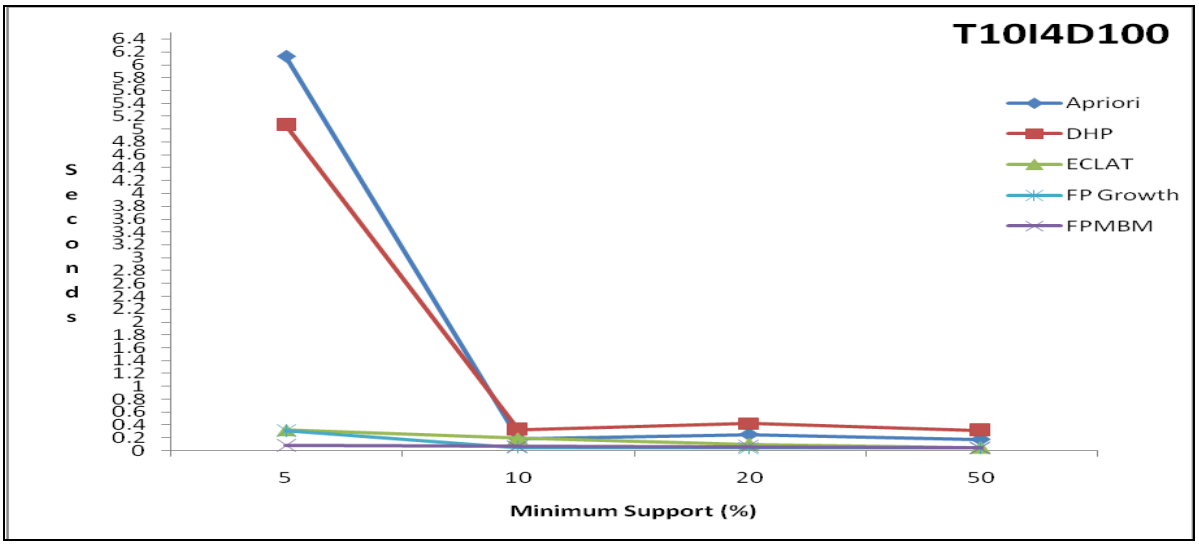

Figure 1. Execution time (in seconds) required by five different algorithms in T10I4D100 dataset with different minimum support threshold.

The dataset in Fig. 1 shows that the idea that the new algorithm runs the greatest on slighter to longer supports with small size dataset. For most supports on the datasets, the new technique has the best result.

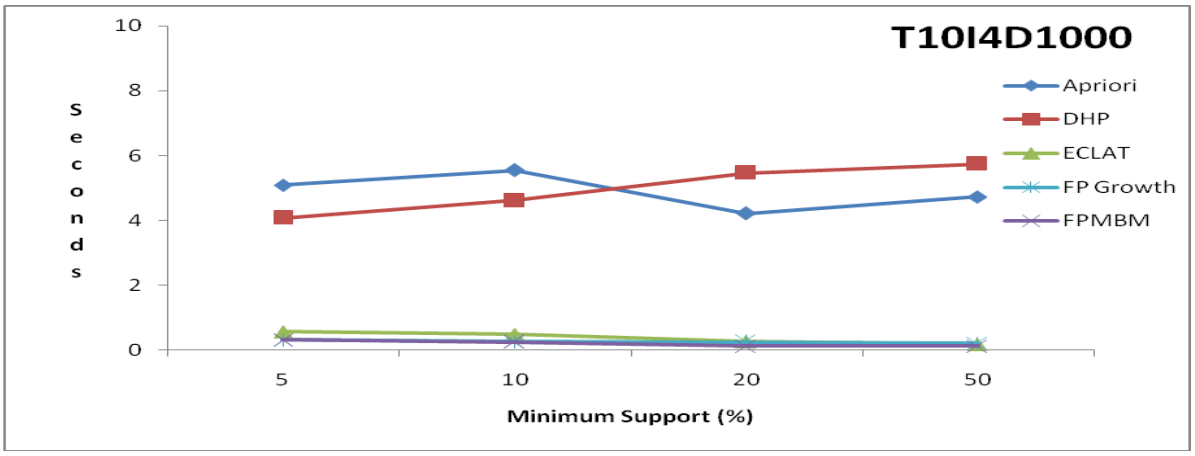

Figure 2. Execution time (in seconds) required by five different algorithms in T10I4D1000 dataset with different minimum support threshold.

Fig. 2 shows the result of computing the new technique with the other four algorithms on dataset. On the above fig, we can find that the new algorithm demonstrates the best result of the four algorithms

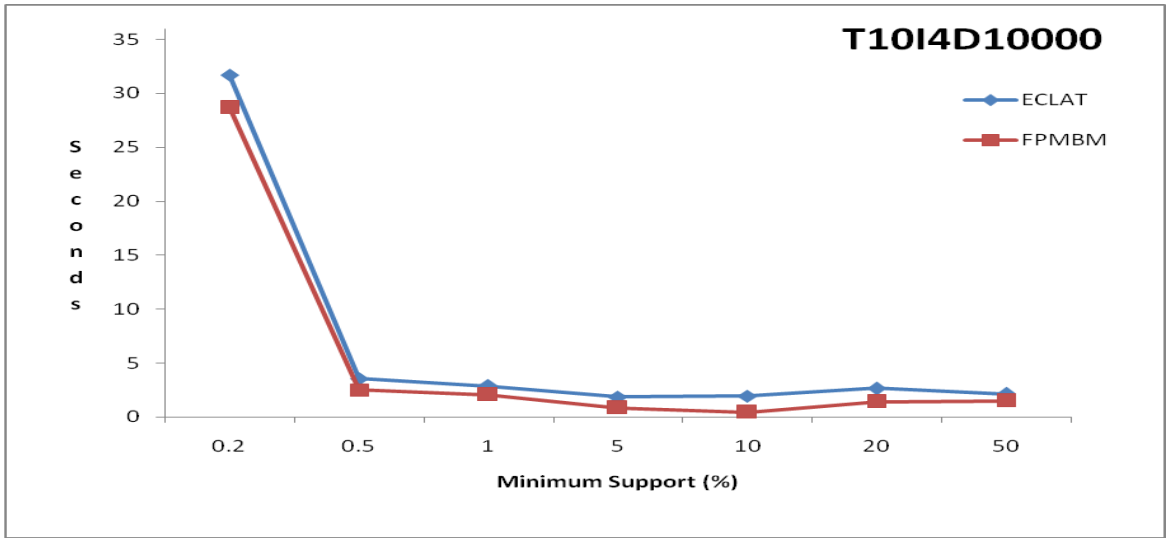

Figure 3. Execution time (in seconds) required by two different algorithms in T10I4D10000 dataset with different minimum support threshold.

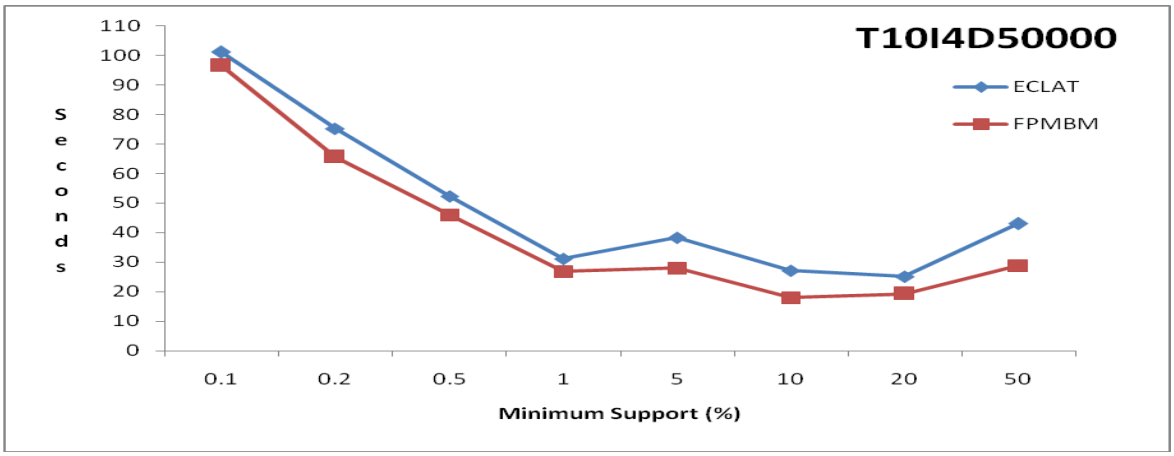

Figure 4. Execution time (in seconds) required by two different algorithms in T10I4D50000 dataset with different minimum support threshold.

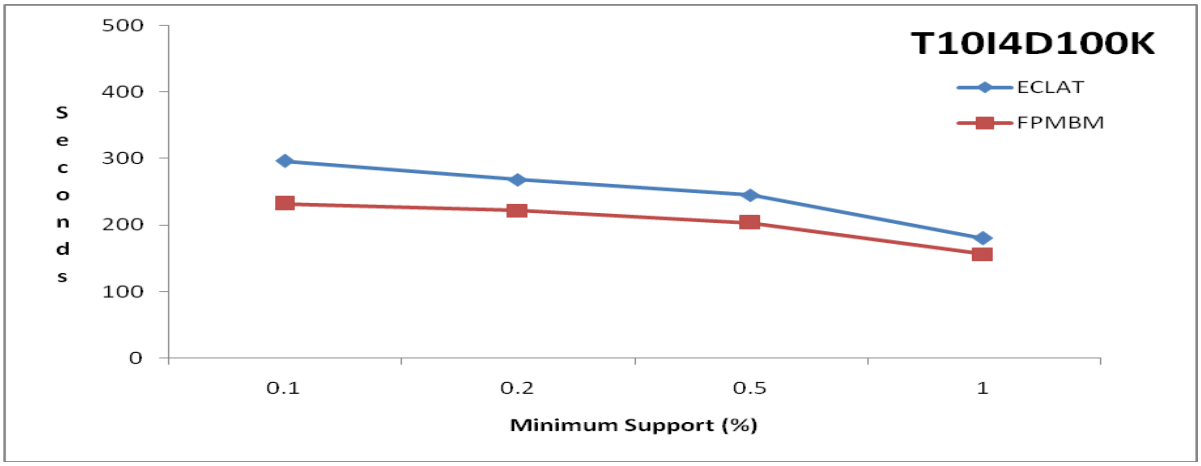

Figure 5. Execution time (in seconds) required by two different algorithms in T10I4D100K dataset with different minimum support threshold.

Fig. 3, 4 and 5 show the computed results of the new algorithm with the ECLAT algorithm on different size of datasets[9]. On the above figures, we can discover that the new technique exhibits the best result compare to ECLAT.

V. Conclusion

Apriori is root for frequent pattern mining loom for newer technique development. But after implementation you can locate some confront like vast amount of database check for creating big pattern and doing support calculation, great number of candidate findings and we can improve these algorithms with effective vertical data and binary value and list for no. iterations based technique for the candidate itemset generation. By following implementation statics discussed above, we can find that FPMBM is very much proficient and scalable frequent pattern mining technique.

References

[1] Data Mining: Concepts and Techniques, Jiawei Han and Micheline Kamber, MORGAN KAUFMANN PUBLISHER, An Imprint of Elsevier

[2] R. Agrawal and S. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases", Proceedings of the 20th International Conference on Very Large Data Bases, September 1994.

[3] J. Park, M. Chen and Philip Yu, "An Effective Hash-Based Algorithm for Mining Association Rules", Proceedings of ACM Special Interest Group of Management of Data, ACM SIGMOD'95, 1995.

[4] Han, Pei & Yin, "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach", Data Mining and Knowledge Discovery, Volume 8, Issue 1 , pp 53-87,2004

[5] M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, " New Algorithms for Fast Discovery of Association Rules", Proc. 3rd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'97, Newport Beach, CA), 283-296 AAAI Press, Menlo Park, CA, USA 1997

[6] Shruti Aggarwal, Ranveer Kaur, "Comparative Study of Various Improved Versions of Apriori Algorithm", International Journal of Engineering Trends and Technology (IJETT) - Volume4Issue4- April 2013

[7] Agrawal, R., T. Imielin´ ski, and A. Swami (1993). Mining association rules between sets of items in large databases. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, SIGMOD '93, New York, NY, USA, pp. 207–216. ACM.

[8] Honglie Yu, Jun Wen, Hongmei Wang, Li Jun (2011). "An Improved Apriori Algorithm Based On the Boolean Matrix and Hadoop". In Procedia Engineering, Volume 15, 2011, CEIS 2011, by SciVerse ScienceDirect.pp.1827–1831.

[9] Synthetic Data for Associations and Sequential Patterns. http://fimi.cs.helsinki.fi.