ISSN:0975-9646

Comparative Analysis of Association Rule Mining Algorithms Based on Performance Survey

K.Vani
Department of Computer Science,Bharathiyar University
Ciombatore,Tamilnadu

***Abstract*-Association rule mining has been focused as a major challenge within the field of data mining in research for over a decade. The research and tremendous progress has been made, ranging from efficient and scalable algorithms for frequent itemset mining in transaction databases to numerous research frontiers. The time required for generating frequent itemsets plays an important role. In this paper includes performance survey of various algorithms and compare those algorithms based on execution time using various datasets and support values. This paper also compared the merits and demerits of ARM algorithms. Finally this paper present which algorithm is suitable for which dataset.**

***Keywords*-Data mining, Association Rule Mining, Apriori, FP-Growth and Eclat.**

I.INTRODUCTION

Data mining has long been an active area of research in databases. The day by day decreasing cost and compactness of storage devices has made it possible to store every transaction of a transactional database. This storage solves two problems first they can access the data any times second this data helps them to find relationship among data items. The problem of finding relationship among different data items was first introduced by agarwalet. al. The solution to this problem can help to enhance the earnings, optimized storage. In this section the researcher introduces the concept of transactional database, database layout, frequent pattern, frequent itemset and candidate itemset. A database is a systematically arranged collection of data, so that it can be retrieved and manipulated easily at a later time. There are different kinds of database, like active database, cloud database, embedded database and transactional database etc, but in this paper the researcher deals with transactional database only. A transactional database is a database in which there is no auto commit. Most modern relational database are the transactional database.

The term data mining or knowledge discovery in database has been adopted for a field of research dealing with the automatic discovery of implicit information or knowledge within the databases. The implicit information within databases, mainly the interesting association relationships among sets of objects that lead to association rules may disclose useful patterns for decision support, financial forecast, marketing policies, even medical diagnosis and many other applications. The problem of mining frequent itemsets arose first as a sub problem of mining association rules. Frequent itemsets play an essential role in many data mining tasks that try to find interesting patterns from databases such as association rules, correlations, sequences, classifiers, clusters and many more of which the mining of association rules is one of the most popular problems. The original motivation for searching association rules came from the need to analyze so called supermarket transaction data, that is, to examine customer behavior in terms of the purchased products. Association rules describe how often items are purchased together. For example, an association rules "beer, chips (80%)" states that four out of five customers that bought beer also bought chips. Such rules can be useful for decisions concerning product pricing, promotions, store layout and many others.

A. Association Rule Mining

Association rule mining [ARM] is the one of the best signed and glowing researched methods of data mining. Association rule mining is a great resolution designed for substitute rule mining, since its objects to realize entirely rules in data and as a result is able to arrange for a whole depiction of associations in a huge dataset. Present area, yet, two most important problems by way of regard towards the association rule generation. At first problem branches formation the rule quantity and excellence problems. Unknown least provision is set as well as high; the rules concerning intermittent substances that can be of interest to resolution makers will not be initiated. Situation least provision low, however, container cause combinatorial explosion. In other words, else several rules are produced regardless of their interestingness33. Several algorithms container be used to realize association rules from data to abstract useful arrays. Apriori algorithm is one of the greatest extensively used and illustrious techniques for discovering association rules. Given a set of relations somewhere each of relatives stays a set of items (itemset), an association rule indicates the form X$\breve{}$∪Y, where X and Y stay itemsets; X and Y are called the body and the head, individually. A rule container be calculated by two processes, entitled confidence and support. A ration, support used for association rule X∪Y is the fraction of relations that have both itemset X and Y between all relations. The assurance for the rule X∪Y is the measurement of connections that enclose an itemset Y in the intermediate of the transactions that contain an itemset X. The sustenance signifies the utility of the exposed rules and the assurance signifies the inevitability of the rules. Generally association rule mining contains following steps:

- The set of candidate k-itemsets is generated by 1-extensions of the large (k -1) itemsets generated in the previous iteration.

- Supports for the candidate k-itemsets are generated by a pass over the database.
- Itemsets that do not have the minimum support are discarded and the remaining itemsets are called large k-itemsets.

This paper gives a performance evolution on different association rule mining algorithms particularly Apriori, FP-growth and Éclat .The algorithms are analysed based on the performance and using various datasets. This paper also gives the comparison of algorithms based on execution time and support value. The paper is organized as follow: Section I gives the detailed introduction on data mining and association rule mining and section II discuss the problem study and section III explain the review of literature and section IV focuses on Association Rule Mining Algorithms and its performance and section V discuss result and analysis and finally section VI explain conclusion.

II. PROBLEM STUDY

A. Need of Frequent Itemset Mining

Studies of Frequent Itemset (or pattern) Mining is acknowledged in the data mining field because of its broad applications in mining association rules, correlations, and graph pattern constraint based on frequent patterns, sequential patterns, and many other data mining tasks. Efficient algorithms for mining frequent itemsets are crucial for mining association rules as well as for many other data mining tasks. The major challenge found in frequent pattern mining is a large number of result patterns. As the minimum threshold becomes lower, an exponentially large number of itemsets are generated. Therefore, pruning unimportant patterns can be done effectively in mining process and that becomes one of the main topics in frequent pattern mining. Consequently, the main aim is to optimize the process of finding patterns which should be efficient, scalable and can detect the important patterns which can be used in various ways.

III. LITERATURE REVIEW

Jochen Hipp et al provided several efficient algorithms that cope up with the popular and computationally expensive task of association rule mining with a comparison of these algorithms concerning efficiency. He proposed that the algorithms show quite similar runtime behavior in their experiments.

Rakesh Aggarwal and Ramakrishnan Srikant presented two new algorithms, Apriori and AprioriTID, for discovering all significant association rules between items in a large database of transactions and compared these algorithms to the previously known algorithms, the AIS and SETM algorithms. They proposed that these algorithms always outperform AIS and SETM.

Christian Borgelt provides efficient implementation of Apriori and Eclat algorithms. Finding frequent item sets in a set of transactions is a popular method for so called market basket analysis, which aims at finding regularities in the shopping behavior of customers of supermarkets, mail-order companies, on-line shops etc. In particular, it tries to identify sets of products that are frequently bought together. The main problem of finding frequent item sets, i.e. item sets that are contained in a user-specified minimum number of transactions, is that there are so many possible sets, which render naive approaches infeasible due to their unacceptable execution time. Among the more sophisticated approaches two algorithms known under the names of Apriori and Eclat are most popular. Both rely on a top down search in the subset lattice of the items. He proposed for free item sets Eclat wins the competition with respect to execution time and it always wins with respect to memory usage. The data set in which it takes lead is for the lowest minimum support value tested, indicating that for lower minimum support values it is the method of choice, while for higher minimum support values its disadvantage is almost negligible. For closed item sets the more efficient filtering gives Apriori a clear edge with respect to execution time. For maximal item sets the picture is less clear. If the number of maximal item sets is high, Apriori wins due to its more efficient filtering, while Eclat wins for a lower number of maximal item sets due to its more efficient search.

Christian Borgelt presented a paper on Recursive Elimination algorithm. He proposed that if a quick and straightforward implementation is desired, it could be the method of choice. Even though its underlying scheme—which is based on deleting items, recursive processing, and reassigning transactions—is very simple and works without complicated data structures, recursive elimination performs surprisingly well.

IV. ASSOCIATION RULE MINING ALGORITHMS AND ITS PERFORMANCE

A. Apriori Algorithm

Agrawal and Srikant (1994) firstly proposed Apriori algorithm. This algorithm is based on Apriori property which states "every sub (k-1)-Itemset of frequent k-Itemset must be frequent". Two main process are executed in Apriori algorithm: one is candidate generation process, in which the support count of the corresponding sensor items is calculated by scanning transactional database and second is large itemset generation, which is generated by pruning those candidate itemsets which has a support count less than minimum threshold. These processes are iteratively repeated until candidate itemsets or large itemsets becomes empty. Original database is scanned first time for the candidate set, consists of one sensor item and there support has counted, then these 1-Itemset candidates are pruned by simply removing those items that has an item count less than user specified threshold (in above case threshold=30%). In second pass database is scanned again to generate 2-Itemset candidates consist of two items, then again pruned to produced large 2-Itemset using Apriori property. According to apriori property every sub 1-Itemset of 2 frequent itemsets must be frequent. This process ends as in fourth scan of database 4- Itemset candidate will be pruned and large itemset will be empty.

TABLE I
PERFORMANCE SURVEY FOR APRIORI ALGORITHM

S.No	Performance Factor	AprioriAlgorithm
1	data structure	array based
2	technique	use apriori property and join and prune method
3	memory utilization	due to large amount of candidate are produced so require large memory space
4	no.of.scans	multiple scan for generate candidate set
5	execution time	execution time is more as time wasted in producing candidates at every time
6	databases	suitable for sparse datasets as well as dense datasets
7	accuracy	less
8	applications	best for closed itemset

TABLE II
PERFORMANCE SURVEY FOR FP-GROWTH ALGORITHM

S.No	Performance Factor	Fp-Growth Algorithm
1	data structure	tree based
2	technique	it constructs conditional frequent pattern tree and conditional pattern base from database which satisfy the minimum support
3	memory utilization	due to compact structure and no candidates generation require less memory
4	no.of.scans	scan the database twice
5	execution time	small than apriori algorithm
6	databases	suitable for large and medium datasets
7	accuracy	more accurate
8	applications	large itemset

B.FP-Growth Algorithm

To break the two drawbacks of Apriori algorithm, FP-growth algorithm is used. FP-growth requires constructing FP-tree. For that, it requires two passes. FP-growth uses divide and conquer strategy. It requires two scans on the database. It first computes a list of frequent items sorted by frequency in descending order (F-List) and during its first database scan. In the second scan, the database is compressed into a FP-tree. This algorithm performs mining on FP-tree recursively. There is a problem of finding frequent itemsets which is converted to searching and constructing trees recursively. The frequent itemsets are generated with only two passes over the database and without any candidate generation process. There are two sub processes of frequent patterns generation process which includes: construction of the FP-tree and generation of the frequent patterns from the FP-tree.FP-tree is constructed over the data-set using 2 passes.

1) *Pass-1:* Scan the information and realize support for every item and discard rare things. Then type frequent things in downward order that is based on their support.By exploitation this order we will build FP-tree, so common Prefixes will be shared.
2) *Pass-2:* Here nodes correspond to things and it'sacounter.FP-growth reads one dealings at a time then maps it to a path. Mounted order is employed, so methods will overlap once transactions share the things.
3)

In this case, counters are incremented. Some pointers are maintained between nodes that contain identical item, by creating on an individual basis coupled lists. The lot of methods that overlap, higher the compression. FP-tree could slot in memory. Finally, frequent itemsets are extracted from the FP-tree.

C.EclatAlgorithm

Eclat algorithms generate frequent items only once. Frequent itemsets are those items which are frequently occur in the database. There are number of algorithms for finding frequent itemsets. Apriori, is basic algorithm for finding frequent itemsets. But it take more time for finding the frequent itemsets, It needs to scan the database again and again which is time consuming process. In this algorithm we need to calculate support and confidence, so eclat algorithm is developed to remove the limitations of Apriori, algorithm. Eclat algorithm uses vertical database. By which it need to scan the database only once. Support is counted in this algorithm. Confidence is not calculated in this algorithm. Steps for parallel Eclat algorithm:

- Divide the database evenly into horizontal partitions among all processes;
- Each process scans its local database partition to collect the counts for all 1-itemsets and 2-itemsets;
- All processes exchange and sum up the local counts to get the global counts of all 1-itemsets and 2-itemsets, and find frequent ones among them;
- Partition frequent 2-itemsets into equivalence classes by prefixes;
- Assign the equivalence classes to processes;
- Each process transforms its local database partition into vertical tid-lists for all frequent 2-itemsets;
- Each process exchanges the local tid-lists with other processes to get the global ones for the assigned equivalence classes;
- For each assigned equivalence class on each process, recursively mine all frequent itemsets by joining pairs of itemsets from the same equivalence class and intersecting their corresponding tid-lists.

TABLE III
PERFORMANCE SURVEY FOR ECLAT ALGORITHM

S.No	Performance Factor	EclatAlgorithm
1	data structure	array based
2	technique	use intersection of transaction ids list for generating candidate itemsets
3	memory utilization	require less amount of memory compare to apriori if itemsets are small in number
4	no.of.scans	continuously scan and update the database
5	execution time	execution time is small then apriori algorithm
6	databases	suitable for medium and dense datasets but not suitable for small datasets
7	accuracy	more accurate
8	applications	best for free itemset

V. RESULT AND DISCUSSION

Based on that performance survey, Apriori algorithm has a poor performance compared than other two association mining algorithms. So the result and discussion take only the two Fp-growth and Eclat algorithms using different dataset applications.

A.Data Set Requirements

For the experiment we have used datasets of different application. These datasets was obtained from the UCI repository of machine learning databases. The Table IV below portraits the characteristics of the datasets selected for the experiment.

TABLE IV
DATASETS USED IN COMPARISON

File Name	Division	Dist/Rand	Records	I/P Columns
adult dataset n48842.c2.num	5	yes	48842	15
census dataset	0	no	48842	14
le recog di06.n20000.c26.num	5	yes	20000	17
mushroom dataset d90.n8124.c2.num	5	yes	8124	23

B.Performance Comparisons

We have conducted a detailed study to assess the performance of FP-growth and Eclat algorithms. The performance of experiment is total execution time for generating itemset. In this comparison also has same dataset with different threshold support values range from 30% to 70%.

The table V shows the execution time for Fp-growth and Eclat algorithms with different support value (threshold) for adult data set.

TABLE V
EXECUTION TIME FOR ALGORITHMS

Support (Threshold) Value	Execution Time In Seconds	
	Fp-Growth	Eclat
30	0.56	0.54
40	0.5	0.49
50	0.49	0.45
60	0.48	0.44
70	0.42	0.4

Figure 1 shows that the execution time for the FP-growth algorithm and Eclat algorithm for adult dataset. The execution time decreased when the support value (threshold) values are increased. Finally we observe that Eclat algorithm produce best performance for adult dataset.

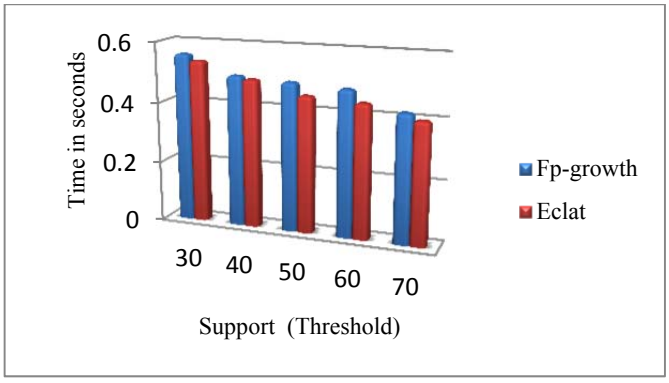

Fig.1 Execution time for adult dataset

The table VI shows the execution time for Fp-growth and Eclat algorithms with different support value (threshold) for Census data set.

TABLE VI
EXECUTION TIME FOR ALGORITHMS

Support (Threshold) Value	Execution Time In Seconds	
	Fp-Growth	Eclat
30	1.21	0.76
40	1.16	0.75
50	0.86	0.72
60	0.73	0.69
70	0.68	0.64

Figure 2 shows that the execution time for the FP-growth algorithm and Eclat algorithm for Census dataset. The execution time decreased when the support value (threshold) values are increased. So, we observe that Eclat algorithm produce best performance for Census dataset.

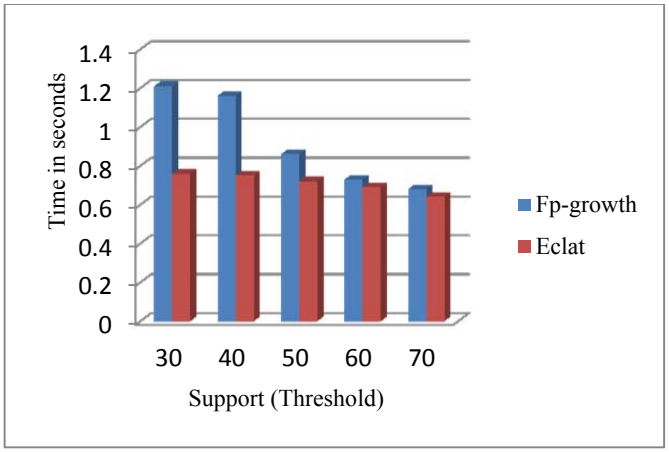

Fig.2 Execution time for census dataset

The table VII shows the execution time for Fp-growth and Eclat algorithms with different support value (threshold) for Letter Recognition data set.

TABLE VII
EXECUTION TIME FOR ALGORITHMS

Support (Threshold) Value	Execution Time In Seconds	
	Fp-Growth	Eclat
30	0.21	0.21
40	0.2	0.21
50	0.18	0.2
60	0.17	0.19
70	0.15	0.17

Figure 3 shows that the execution time for the FP-growth algorithm and Eclat algorithm for Letter Recognition dataset. The execution time decreased when the support value (threshold) values are increased. So, we observe that Fp-growth algorithm produce best performance for Letter Recognition dataset.

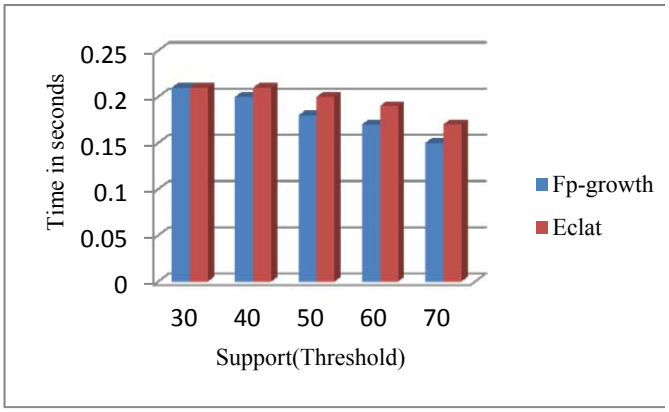

Fig.3 Execution time for letter recognition dataset

The table VIII shows the execution time for Fp-growth and Eclat algorithms with different support value (threshold) for Mushroom data set.

TABLE VIII
EXECUTION TIME FOR ALGORITHMS

Support (Threshold) Value	Execution Time In Seconds	
	Fp-Growth	Eclat
30	0.13	0.11
40	0.11	0.11
50	0.09	0.09
60	0.08	0.09
70	0.08	0.08

Figure 4 shows that the execution time for the FP-growth algorithm and Eclat algorithm for Mushroom dataset. The execution time decreased when the support value (threshold) values are increased. So, we observe that both Fp-growth and Eclat algorithms are produce best performance for Mushroom dataset.

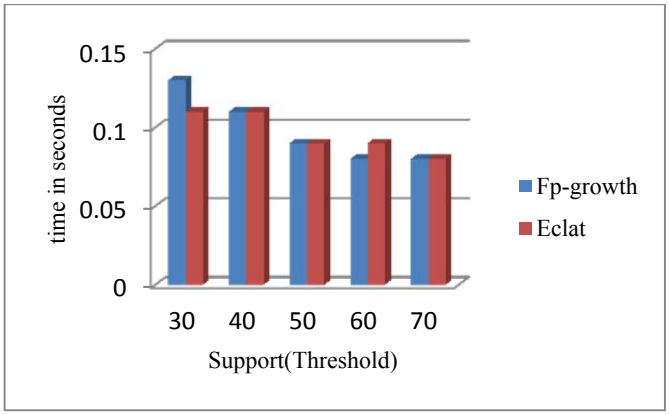

Fig.Execution time for mushroom dataset

VI. CONCLUSION

A comparison framework has been developed to allow the flexible comparison of existing and new frequent itemset mining algorithms that conform to the defined algorithm performance. Using this framework the paper present comparative analysis based on performance survey of three iterative algorithms like apriori, Fp-growth and Eclat algorithms. The performance survey include factors like data structure , technique, memory utilization, no. of scans, execution time, databases, accuracy and applications. Based on that survey the apriori algorithm provides poor performance. So the comparative analyses take only Fp-growth and Eclat algorithms.

In this work, an in-depth analysis of few algorithms is done which made a significant contribution to the search of improving the efficiency of frequent itemset mining. By comparing these two algorithms based on support value (threshold) and execution time. The support (threshold) values are increased the execution time was decreased.

In this paper comparison framework has been developed using various dataset like adult, census, letter recognition and mushroom. The datasets are compare those classical frequent item set mining algorithms. Finally this paper concludes which algorithm provides better performance for which datasets. The below table IX explain the dataset for suitable algorithm.

TABLE IX
DATASET FOR ALGORITHMS

S.No	Dataset	Algorithm
1	adult	eclat
2	census	eclat
3	letter recognition	fp-growth
4	mushroom	eclat and fp-growth

ACNOWLEDEMENT

I thank God Almighty for giving me good dynamism and wisdom to do my research paper and complete it successfully. It is my immense pleasure deep satisfaction and gratitude to acknowledge the contributors towards the successful completion of this paper.

Guidance, Help and Motivation are the vital bricks in the construction of a research work. I would like to extend my gratitude to all individual who made me to complete this research work in a successful manner.

I would like to express my deep sense of gratitude and sincere thanks to my guide **Mrs.D.UmaMaheswari MSc.,M.Phil.,** Assistant Professor , Department of Computer Science, Vidyasagar College of Arts and Science, Udumalpet, for his valuable suggestions and timely guidance, which paved way to streamline and who is behind the success of this research. It would not have been possible for me to proceed, but her motivation, expert guidance and encouragement provided throughout the successful completion of the dissertation.

I take this opportunity to express my sincere gratitude and thanks to my beloved Parents, Husband, Friends and Research Scholars in Department of Computer science, for their moral support, inspiration, and encouragement and kind of co-operation throughout the research.

REFERENCES

[1] C. Borgelt. *An Implementation of the FP- growth Algorithm*.Proc. Workshop Open Software for Data Mining (OSDM'05 at KDD'05, Chicago, IL), 1–5.ACMPress, New York, NY, USA 2005.

[2] M. Zaki, S. Parthasarathy, M. Ogihara, and W. Li. *NewAlgorithms for Fast Discovery of Association Rules*. Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD'97), 283–296. AAAI Press, Menlo Park, CA, USA 1997

[3] 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.

[4] J. Han, J. Pei, Y. Yin, and R. Mao. *Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Mining and Knowledge Discovery*, 2003.

[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] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A.I. Verkamo.Fast discovery of association rules. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, pages 307–328. MIT Press, 1996.

[7] C.L. Blake and C.J. Merz. *UCI Repository of Machine Learning Databases*. Dept. of Information and Computer Science,University of California at Irvine,CA, USA1998 http://www.ics.uci.edu/˜mlearn/MLRepository.html- 1998.

[8] Pramod S., and Vyas O.P., *Survey on Frequent Item set Mining Algorithms*, In Proc. International Journal of Computer Applications (0975-8887), 1(15), 86–91 (2010)

[9] R. Agrawal, T. Imielienski, and A. Swami.*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.

[10] C. Borgelt. SaM: *Simple Algorithms for Frequent Item Set Mining*. IFSA/EUSFLAT 2009 conference- 2009.

[11] J. Han, and M. Kamber, 2000.*Data Mining Concepts and Techniques*. Morgan Kanufmann.