iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** II **Month of publication:** February 2018

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887
Volume 6 Issue II, February 2018- Available at www.ijraset.com

Analysis of Association Rule Mining Algorithms

Divya Gautam
Amity University, Madhya Pradesh, Gwalior

Abstract: Association rule mining is an important field of data mining. It is very expensive still it is very popular because of its importance and usefulness. The performance of algorithm is very efficient and effective to data mining results. Frequent Item sets mining has a significant role in association rule mining. Apriori algorithm and FP- growth algorithm are the famous algorithms to solve the problem of frequent item sets. A programmer should know all the possibilities which are important to know to make the task given to them efficient and effective. This paper discusses and compares various algorithms for association rule mining.

Keywords: Data mining rules, Apriori algorithm, SETM.

I. INTRODUCTION

A. Data Mining

Data mining is an important process in which brilliant means are used to mine or extract useful data. It is an integrative subfield of computer science. The main aim of this process is to extract useful data from the set of data and convert the data into a form which is understandable and meaningful or which can be use further in an easy way. The data mining process is one of the investigation step of the “knowledge discovery in databases" process, or we can say “KDD”.

The term called data mining came into view about 1990 in the society of database. A data scientist named Gregory Piatetsky-Shapiro invented the term "knowledge discovery in databases" for the earliest meeting on the similar issue [KDD-1989].

Data mining is most useful in the following fields:

Market scrutiny and Management

Corporate scrutiny & Risk Management

Fraud recognition Besides these, data mining can also be used in the fields of manufacture power, customer retention, science exploration, games, astrology, and social media.

B. Association Rule Mining

Association rule mining is a method which depends on machine learning method for generating attractive
relationships among the variables in huge databases. It is projected to recognize physically powerful conventions revealed in databases using a few procedures of interestingness. Depends on the perception of powerful regulations, Rakesh Agrawal, Tomasz Imieliński and Arun Swami established association rules for generating laws among goods in large-scale transaction data taken by the system called point-of-sale (POS) in big or supermarkets. For instance,a ruling {onion, potatoes}⇒{burger} originates in the data of sales of a supermarket would signify that if a consumer purchases onions and potatoes mutually, then he is willingly to also purchase hamburger meat.

This data can be taken as the origin for making judgments regarding marketing deed and behavior like, increasing pricing or good selling. In accumulation to the previous instance from market basket examination association rules are engaged in many relevance fields like Web usage extraction, intrusion detection, spontaneous generation, and bioinformatics. On the contrary in the midst of sequence extraction, association rule learning normally does not regard as a sort of substance either within a matter or across transactions.

The association rules concept was proposed mainly because of the article[1993] given by Agrawal et al, that has achieved greater than 18,000 records. It is now use in every fields as it is very efficient and effective.

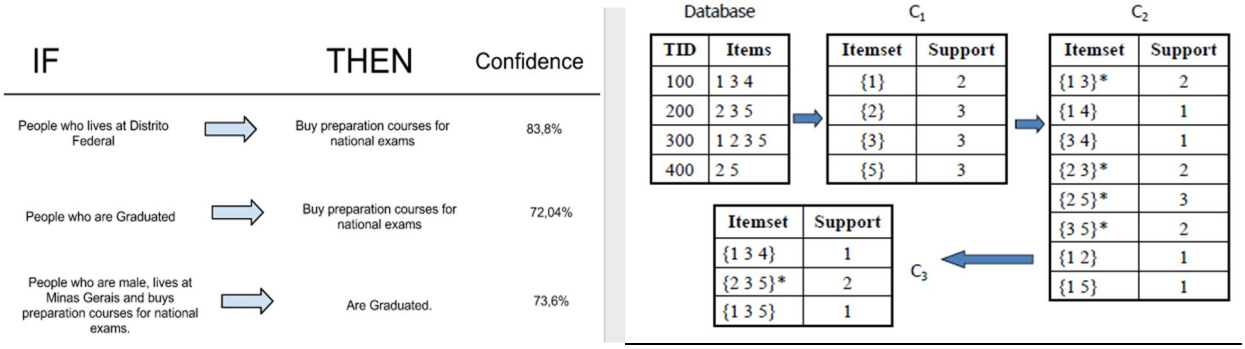


Table of Content

For understanding the data mining and association rules for data mining I have explore some papers and one tutorial also which are given below:-

II. METHODOLOGY

The association rules are basically necessary to assure a user-specified least support and least confidence at the similar instant. The two steps of association rules are given below:

A least support threshold is useful to discover every numerous database itemsets. A least confidence limit(constraint) is useful to the numerous itemsets with the intention of making rules. Since the later one is straight forward, the former one requires additional awareness. It is complex to find all numerous database itemsets since it contains searching every probable itemset or item combinations. The set of probable itemsets is the power place over 'I' and have size 2^n-1 (without unfilled set which is not an applicable itemset). Even though the power-set size increases exponentially in the amount of items 'N' in 'I', well-organized investigate is probable use the "downward-closure property" of support (as well known as anti-monotonicity) that assurance that for a numerous itemsets, all the subsets are too common and hence occasional itemsets can't be a subset of a regular itemsets. By taking advantage of the property, well-organized algorithms (such as, Apriori and Eclat) can discover every single one numerous itemset.

III. ALGORITHMS

In this part, nearly all algorithms are used to recognize huge itemset can be divided into two parts:-

A. Sequential and parallel.

In normal forms, it is understood that the itemset is recognized and accumulated like treasury arranged (depends on the name of item). This arrangement provides a logical mode where itemset can be discovered and calculated. It is the customary move with the sequential algorithms. In contrast, parallel algorithms focal point is how to parallelize the assignment of ruling huge itemset.
We talk about presentation of the algorithms and come across through the data structures used.

B. Various Algorithms

1) *AIS:* The algorithm 'AIS' was the initial available algorithm urbanized to make the entire huge item sets in a transactions database [Agrawal1993]. This is focused on the development of database with obligatory functionality to practice conclusion support queries. The generation of qualitative rules was the target of this algorithm. Only one item is restricted in the consequent by this technique. That is to say, the form of association rules is " X⇒Ij | α", in which 'X' is called set of items and 'Ij' is called single item in the field 'I', and 'α' is the rule's confidence.
2) *SETM:* The SETM algorithm was discovered to use the SQL to evaluate the bulky itemset [Srikant 1996b] in Houtsma in1995. In this algorithm, all members of the set bulky itemset, 'Lk', is in the form of <TID, itemsets>in which 'TID' is the only one of its kind identifier of a transaction. In the same way, all members of the set of candidate itemset, 'Ck', is in the form of <TID, itemsest>. Same as the former algorithm i.e. AIS, the SETM algorithm creates several passes on the database. In the initial pass, it calculates the support of item individually and concludes that which of them are great and regular in the database. aAter that, it discovers the applicant itemset by expanding huge itemset of the last pass. Additionally, the SETM algorithm reminds the TIDs of the discovering transactions with the applicant itemset.
3) Apriori: This algorithm was proposed by Agrawal in1994 is an immense accomplishment in the mining association rules history given by "Cheung" in 1996. This is very well known algorithm. The property that any subset of a huge itemsets must be

a huge item sets is used by this technique. As well, this is implicit that items in item sets are treasury arranged. The original and basic differences of the algorithm from the former two algorithms (AIS and SETM algorithms) are the approach of discovering applicant itemset and the selection of applicant itemset for counting. As said before, that in both the AIS and SETM algorithms, the ordinary itemset among huge itemset of the last pass and objects of a transaction are found. This algorithm is also used for the large sets of data items

4) Apriori-TID: As said before, Apriori examines the whole database from all pass to count support. Examining of the whole database might not be essential in every passes. Depends on this estimate**,** the Agrawal wished-for another algorithm called Apriori-TID which was given in 1994. Alike to Apriori, Apriori-TID utilizes the Apriori's applicant generating purpose to find out applicant itemset earlier than the establishment of the pass. The chief difference between Apriori and Apriori TID algorithm is that the later does not utilize the database for counting support after the primary pass. Slightly, this uses a programming of the applicant itemset used in the last pass represented by '*Ck*' .

5) *Apriori-Hybrid:* The Apriori-hybrid algorithm depends on the thought that this is not essential to use the similar algorithm in every passes on data. As declared in [Agrawal1994], Apriori algorithm has better presentation in former passes, and Apiori-TID outperforms Apriori in afterward passes. Depends on the investigational clarification, the Apriori-Hybrid method was generated which utilizes Apriori in the first passes and changes to Apriori-TID while it supposes that the set *Ck* at the closing stage of the pass will fit in memory. As a result, an evaluation of *Ck* at the closing stage of every pass is essential. As well, there is a price contribution of changing from Apriori to Apriori-TID algorithm. The presentation of this method was also checked by performing experiments for great dataset. It was experienced that Apriori-Hybrid executes better than Apriori algorithm excluding in the case when the changing occurs at the very end of the passes [Srikant1996b].

6) *Off-line Candidate Determination (OCD):* The OCD method is projected in [Mannila1994] depends on the initiative that small samples are generally good for discovering huge itemset. The OCD method uses the combinatorial examination result of the data achieved from last passes to eradicate preventable applicant set. If a subset $Y \subseteq I$ is occasional, at least (1-s) of the transactions should be examined where 's' is called the support threshold. Hence, for little values of s, approximately the whole relationship ought to be examined. It is understandable that if database is extremely big, it is significant to formulate as few passes over the data as achievable. OCD pursues a unlike loom from AIS algorithm to find out applicant set. OCD uses all accessible data from the last passes to snip applicant set among the passes by maintain the pass as easy as achievable. It constructs a set 'Lk' as a group of every huge item sets of size k. applicant set Ck+1 has those sets having size (k+1) that can probably be in Lk+1, given the huge itemset of Lk.

7) *Partitioning:* PARTITION algorithm [Savasere1995] diminishes the amount of database scans to 2. In this algorithm database is partitioned into small database so that it can be easily hold in memory. Suppose the partitions of the database are D1, D2, ..., Dn. In the initial examine, it discovers the local huge itemset in all partition Di ($1 \le i \le n$), i.e. $\{X \mid X.\ count \ge s \times |Di|\}$. The local huge itemset, Li, can be established by use of a level-wise algorithm for example Apriori. Because each partition can well in the main memory, there is no supplementary disk I/O for all partition after loading the partition. In the next examine, it uses the possessions that huge item sets in the entire database should be locally big in at least one partition of the database. A after that the combination of the local big itemset discovered in every partition are used as the applicants and are calculated through the entire database to discover every big itemset.

8) *Sampling:* Sampling algorithm [Toivonen1996] diminishes the amount of database examines to one in the best case and two in the worst. A sample that is well in the central memory is initial pinched from the database. The set of huge itemset is then originated from this sample with a level-wise algorithm for instance Apriori. Suppose the set of huge itemset in the sample is 'PL', that is used as a set of credible huge itemset and used to produce applicants which are confirmed against the entire database . The applicants are created by using the negative border function, BD−, to PL. So the applicants are "BD−(PL)" PL. The negative border function is a simplification of the Apriori_gen function in Apriori. If all itemset in 'PL' are of the similar size, then BD−(PL) = apriori_gen(PL). The dissimilarity lies in the negative border which can be useful to a set of item sets of dissimilar sizes, when the apriori_gen() function only applies to a only size. After the applicants are created, the entire database is examined once to establish the calculation of the applicants.

9) CARMA: The algorithm CARMA (Continuous Association Rule Mining Algorithm) [Hidb1999] conveys the calculation of huge itemset online. Being online, CARMA demonstrates the present association rules to the client and permits the client to modify the constraints, least support and least confidence, at any transaction in the initial examine of the database. It requires maximum 2 database examine. It is similar to DIC. But CARMA produces the itemset on the fly from the transactions. After understanding each transaction, it first increases the amount of the itemset that are subset of the transaction. Next it produces

fresh itemset from the transaction, if every instantaneous subsets of the itemset are presently potentially big w.r.to the recent least support and the database part that is examined. For extra precise forecast of whether the itemsets are potentially huge, it computes an superior hurdle for the amount of the itemsets, that is the sum of its recent amount and the approximation of the happening earlier than the itemsets are produces. The approximation of the happening (are known as maximum misses) is calculated when the itemset is initial gener**.**

IV. COMPARISON OF ALGORITHMS

Here evaluation of algorithms depends upon a number of metrics. Space necessities are expected by appearing at the maximum number of applicants being calculated while any examine of the database. The time necessities can be guesstimate by including the maximum number of database examines desired (I/O approximation) and the maximum number of evaluation procedures (CPU approximation). Because mainly of the transaction databases are accumulated on derived disks and I/O slide is more significant than CPU slide, we center on the number of examines in the whole database. Apparently, the worst case happens while every transaction in the database have every part of items. The number of huge itemset is 2m. In level-wise methods (such as AIS, SETM, Apriori), every huge itemsets in L1 are achieved in the initial examine of the database. Likewise, every huge itemset in L2 are acquired in the following examine, and so on. The itemsets in Lm are obtained in the mth examine. Every algorithm come to an end when no supplementary entries in the huge itemset is created, so an additional examine is required. Thus, the whole database is examined at most (m+1) times. Here it is summoned up that Apriori-TID examines the whole database in the initial pass. Then it uses *Ck* slightly than the whole database in the (k+1)th pass. Yet, it does not help at all in the worst case. This is because *Ck* will hold all of the transactions along with the items in the whole process. On the contrary the OCD method examine the whole database only once at the commencement of the algorithm to get great itemset in L1. Afterwards, OCD and Sampling use only a part of the whole database and the data achieved in the initial pass to discover the applicant itemset of Lk in which $1<k\leq m$. In the succeeding examine they calculate support of each applicant itemset. Thus, there will be 2 examines in the worst case specified sufficient main memory. The PARTITION method also reduces the I/O slide by dropping the amount of database examines to 2. aAlso, CARMA needs at most 2 database examines. The righteousness of an algorithm based on the accurateness of the number of “true” applicants it extend. As we have stated before, every algorithms use huge itemset of last pass(es) to create applicant set. Huge itemset of last itemset are fetch into the central memory to create applicant itemset. Again, applicant itemsets are required to be in the central memory to attain their support counts. Since sufficient memory may not be accessible, dissimilar algorithms suggest dissimilar kinds of buffer managing and storage structures.

V. CONCLUSION

This paper presented one of the most important part of the data mining to find interesting relationships among data of large dataset items. It has been observed that the two techniques are there to find the relations or to establish the relationships among the data items:- by using numerous(frequent) itemsets, in which items are represented that are commonly show in the data collectively. And the next one is the association rules in which if -> then relationship is implied between items.

Although it is very good, efficient and effective still there are many problems with it. Since it is a computing process hence it is very expensive so it is not suitable for the small organization. It can be very consuming task while finding different combinations of items. So we should be familiar to each method or algorithm so that it will take less time to process and we get the accurate result in very less time. In all the algorithms Apriori algorithm seems to be very easy and effective. It is easy to understand and get the desired result. This is one who has the tendency to diminish the number of sets that are checked against the dataset. We use the algorithms by using two two components:- support measure and confidence measure. By combining both the measures association rules are generated. Yes, it is true that it can’t be possible, the Apriori algorithm is good in all prospects. There is a problem with algorithm is that it needs to scan the data set each time. It is not bad for the minimum date sets but it is very bad for the very huge data set. Since it scan the data set each time, it will increase the time and hence speed will decrease of finding the frequent itemsets. To get rid of this problem many algorithm are proposed. Each algorithm is good in it’s way. The alternative method is discovered to eradicate the issues happening with the Apriori algorithm, the hybrid Apriori algorithm. This algorithm scans the whole dataset twice only which decrease the time and obviously increase the speed. This algorithm can led to better performance.

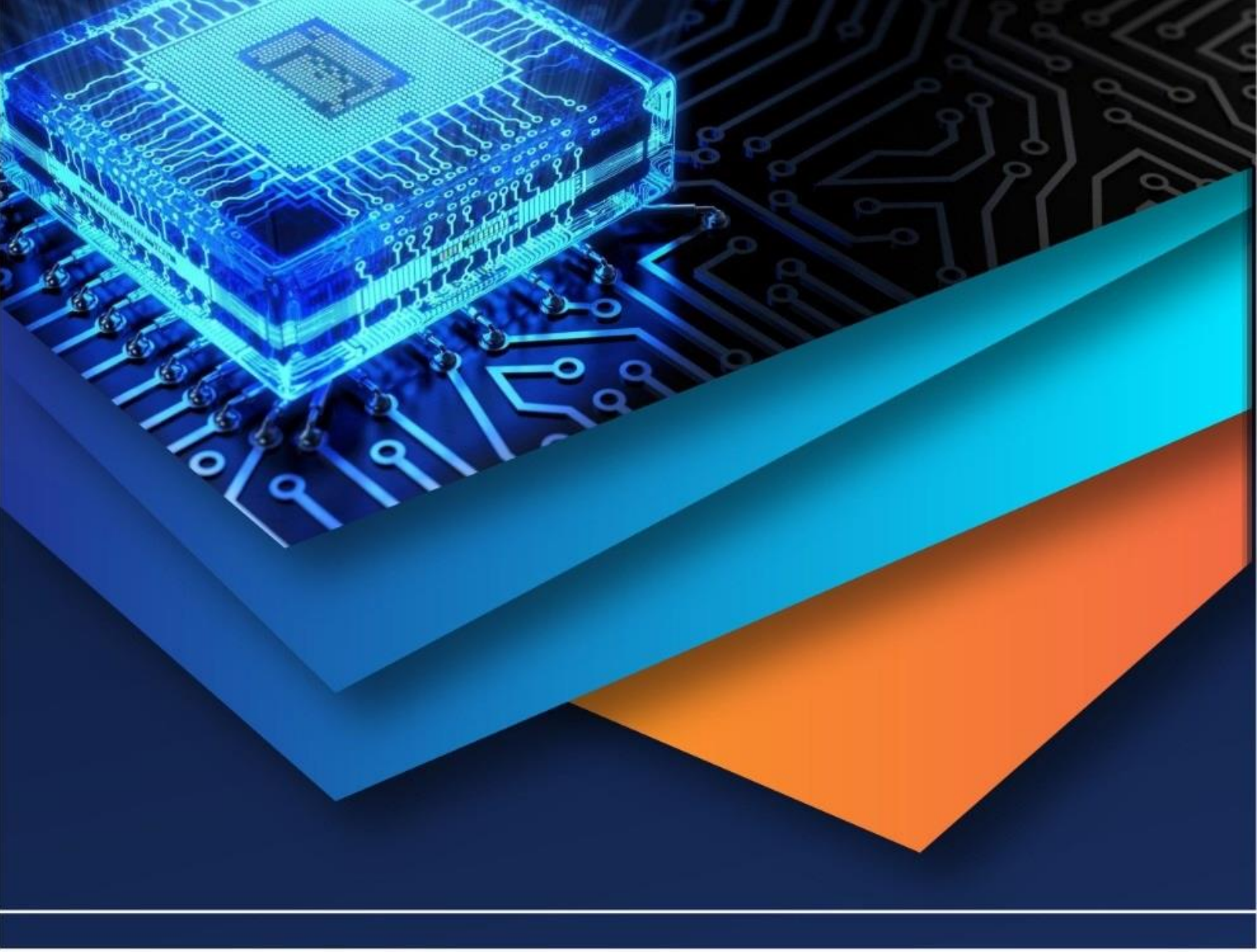
This paper, does not covered all the algorithms. As there are many algorithms it is not possible to explain that all in one paper. But here we can see an overview on the algorithms. The algorithms FP-Growth algorithm, is one the best algorithm which minimize the time to extract or maximize the speed.

Here only sequential algorithms are explained. Parallel and distributed algorithms are also the part of basic algorithms. There so many parallel and distributed algorithms. CD(in 1996), PDM(in 1995), DMA(1996), and CCPD(in 1996) are the data parallelism algorithm and DD(in 1996), IDD(in 1997), HPA(in 1996) and PAR(in 1997) are the task parallelism algorithm. These algorithms are equally important as sequential algorithm.

In today's scenario we are using so many algorithms to manage our data. Algorithms are getting updated and make our work easy. But the set of data or it's length is keep increasing day by day. If it will continue then it will become difficult to manage the data set with these algorithms. Not only these algorithms are used to extract data but there are also many other algorithms as I have discussed above. As our needs keeps on increasing, new algorithms with new upgrades will also take place as the technology will take place. Hence it is necessary to establish new or upgraded algorithms to maintain the whole data set.

REFERENCES

[1] Margaret H. Dunham , Yongqiao Xiao , Le Gruenwald , Zahid Hossain "A Survey Of Association Rules"
[2] Charu C. Aggarwal, Zheng Sun, and Philip S. Yu, Online Algorithms for Finding Profile Association Rules, Proceedings of the ACM CIKM Conference, 1998, pp 86- 95.
[3] C. C. Aggarwal, J. L. Wolf, P. S. Yu, and M. Epelman, Online Generation of Profile Association Rules, Proceedings of the International conference on Knowledge Discovery and Data Mining, August 1998.
[4] Charu C. Aggarwal, and Philip S. Yu, A New Framework for Itemset Generation, Principles of Database Systems (PODS) 1998, Seattle, WA.
[5] Rakesh Agrawal, Tomasz Imielinski, and Arun N. Swami, Mining Association Rules Between Sets of Items in Large Databases, Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, pp. 207-216, Washington, D.C., May 1993. 54
[6] Rakesh Agrawal, Tomasz Imielinski and Arun N. Swami", Data Mining: A Performance perspective, IEEE Transactions on Knowledge and Data Engineering, Vol. 5, No. 6, December 1993, pp. 914-925.
[7] Rakesh Agrawal and Ramakrishnan Srikant, Fast Algorithms for Mining Association Rules in Large Databases, Proceedings of the Twentieth International Conference on Very Large Databases, pp. 487-499, Santiago, Chile, 1994. Rakesh Agrawal and Ramakrishnan Srikant, Mining Sequential Patterns, Proceedings of the 11th IEEE International Conference on Data Engineering, Taipei, Taiwan, March 1995, IEEE Computer Society Press. Rakesh Agrawal, Heikki Mannila, Ramakrishnan Srikant, Hannu Toivonen, and A. Inkeri Verkamo, Fast Discovery of Association Rules, In Usama M. Fayyad, Gregory PiatetskyShapiro, Padhraic Smyth, and Ramasamy Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, pp. 307-328, Menlo Park, CA, 1996. AAAI Press.
[8] Rakesh Agrawal and John C. Shafer, Parallel Mining of Association Rules, IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 6, pp. 962-969, December 1996. Roberto J. Bayardo Jr., Rakesh Agrawal, Dimitris Gunopulos, Constarint-Based Rule Mining in Large, Dense Databases, Proceedings of the 15th International Conference on Data Engineering, 23-26 March 1999, Sydney, Australia, pp.188-197
[9] Sergey Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur, Dynamic Itemset Counting and Implication Rules for Market Basket Data, Proceedings of the ACM SIGMOD Conference, pp. 255-264, 1997.
[10] Sergey Brin, Rajeev Motwani, and Craig Silverstein, Beyond Market Baskets: Generalizing Association Rules to Correlations, Proceedings of the ACM SIGMOD Conference, pp. 265-276, 1997.
[11] Ilker Cengiz, Mining Association Rules, Bilkent University, Department of Computer Engineering and Information Sciences, Ankara, Turkey, 1997, URL: http://www.cs.bilkent.edu.tr/~icegiz/datamone/mining.html.
[12] Jaturon Chattratichat, John Darlington, Moustafa Ghanem, and et. al, Large Scale Data Mining: Challenges and Responses, Proceedings of the 3th International Conference on Knowledge Discovery and Data Mining, pp. 143-146, August 1997.
[13] Ming-Syan Chen, Jiawei Han and Philip S. Yu, Data Mining: An Overview from a Database Perspective, IEEE Transactions on Knowledge and Data Engineering, Vol. 8, No. 6, pp. 866-883, 1996.
[14] Data mining tutorials-www.tutorialpoints.com.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY