ISSN(Online) : 2320-9801
ISSN (Print) :  2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 4, April 2016

# Incremental Association Data Mining Using Quick sort and Searching Approach

Disha Vaghasiya[1], Dr.Vipul Vekariya[2]

PG Student, Department of Computer Engineering, Noble Engineering College, Junagadh, Gujarat, India[1]

HOD, Department of CE, Noble Engineering College, Junagadh, Gujarat, India[2]

**ABSTRACT**: Association rule is very effective and useful in the area of data mining. In dynamic and functional databases new transactions are adjoined as time advances.so, this may introduce new association rules and some existing and current rules would become inadmissible. So, to maintain association rules for dynamic database is the vital issue. So, in this paper I have new arrival of incremental algorithm for frequent item set is proposed to deal with this problem. This approach is based on quick sort algorithm and proposed algorithm uses maximum support court of 1-itemset which is obtained from previous mining to calculate in frequent item set called promising item set. When new transactions are being added to original database, the algorithm take less no. of time scan the original database so this paper algorithm has a simulation result that shows good performance. Enhance, this paper also includes new approach searching algorithm with incremental association.

**KEYWORDS**: Frequent Itemset, Incremental Mining, Quick sort.

## I. INTRODUCTION

Frequent item set finding with an incremental data is an important subject in lots of data mining applications, such as discovery of association rules, correlations, sequential rules and episodes. Data mining has a great significant in real world applications. Day by day, database size is increasing due to increasing use of large amount of data requires high computation for different applications so that the importance of data mining has grown rapidly. One of the primary applications of data mining is association rule mining technique. Association data rule mining is the technique which finds correlation between two item sets.

For static database traditional rule mining can work efficiently but for dynamic database traditional association rule mining is a trade-offs, for dynamic database incremental association rule mining technique is used. It's a technique in which afterwards some period of time new transaction are added so new calculation must be required and it may be possible that some old rule can be obsolete. Association rule mining [10], which is one kind of data mining algorithms, becomes now a days more and more popular these years. It tends to identify strong rules between no less than two items in database through different measures of interestingness. However, when it comes to large data, related algorithms are not mature efficient and need further research. In a practical condition, database is updated periodically and threshold value often changes with needs of rule mining.

The goal and purpose of this work is to solve the efficient updating problem of association rules is that after a number of new records have been added to a database and finding frequent itemset with updating.

Our new approach introduces a promising frequent itemset for an infrequent itemset it may be capable of being a frequent itemset after numbers of new records have been added to the database. This can be reduced number of times to scan original database. As a result, the algorithm has the execution time faster than that of previous methods with searching technique.

## II. RELATED WORK

An imposing algorithm for association rule mining is Apriori[4]. Apriori algorithm computes frequent itemsets in the large database through numerous iterations based on a prior knowledge. Each has 2 steps. For every iteration with 2

DOI: 10.15680/IJIRCCE.2016. 0404286

steps, processes are joining and prune step. For an each frequent itemset, its support must be higher than a user-specified minimum support of threshold. Association rule can be discovered based on frequent item sets that should be higher than user-specified minimum confidence. FP-Growth algorithm [6] solved the “scan” problem by building the FP-tree to mine the frequent itemsets.

AprioriTidList algorithm [8] was presented by Liu Han-bing in 2011 in that Main features of ATLUP algorithm are that: frequent item sets of new transaction in database are produced by AprioriTidList algorithm, and candidate itemsets that are classified and pruned in effective ways. So, the time of scanning by gone sand new database is reduced to once, efficiency of updating association rules is being improved.

Extracting association rules is a fundamental activity among the all the data mining tasks. Marghjny and mohammed said about apriori count table [2] in which first The main thing of (association rules) is to searchout for interesting relationship among the items in given database and shows it in a rule form, i.e. A⇒B. the well-known constraints are Sminimum thresholds on support(min_sup) and confidence(min_conf). This rule A ⇒B holds with support s if s% of transactions in D contain. A ⇒ B. Rules that have as greater than the user-specified support is said to have min_sup. The A ⇒ B holds it with confidence c if c% of the transactions in D that contain A and also contain B. these Rules which have a c greater than a user-specified confidence is said to have min_conf.

It is well known that transaction T is said to the contain A if and only if A ⊆ T: In moreover association rules are an canotation of the form A ⇒ B, where I ⊃ A; B ⊃ I; and A ∩ B = ϕ where I = {i1, i2, . . ., im} be a set of items. The rule A ⇒ B takes in the transaction set D, where D is the task-relevant data, be the set of database transactions where each of transaction T is a set of items whereas T ⊆ I with support s, where s is a percentage of transactions in D that contain (A ∪ B) (the union of sets A and B, or both A and B). This is held to be the probability, P (A ∪ B). The rule A⇒ B has a confidence c in the transaction set D, where c is a percentage of transactions in D contains A that also contain B. This is held to be the conditional probability, P (B|A). That is,

$$\mathbf{Sup\ (A \Rightarrow B) = P\ }(\boldsymbol{A} \cup \boldsymbol{B})$$

$$\mathbf{Conf(A \Rightarrow B\ ) = P(B|A)}$$

The given Rules satisfy both a minimum support and a minimum confidence thresholdare called strong. A set of item is said to be an itemset.An item set contains k items is the k-itemset. The set of {milk, bread} is 2-itemset. The occurrences frequency ofan item set is the number of transactions that contains theitemset. This is also known simply, as frequency,support count, or can say count of the itemset. Note that the itemsetsupport defined in the given Equation is sometimes referred to asrelative support, whereas the occurrence frequency iscalled the absolute support. If relative support of Item set satisfies a pre-specified minimum support count threshold then I will be frequent itemset. The set of frequent k-itemsets is usually denoted by Lk. From that, we have

$$\boldsymbol{conf}(\boldsymbol{X} \Rightarrow \boldsymbol{Y}) = \boldsymbol{P}(\boldsymbol{Y}|\boldsymbol{X}) = \frac{\mathbf{sup}\ (\boldsymbol{X} \cup \boldsymbol{Y})}{\mathbf{sup}\ (\boldsymbol{X})}$$

This equation shows the confidence of rule (X⇒Y) which can be easily derived from the support counts of X & X ∪ Y. It is, once support count of X ∪ Y, and X & Y are found, it is easy to derive the corresponding association rules X ⇒Y and Y ⇒X and check whether it is strong. Thus the problem of mining association rules that can be reduced of mining frequent itemsets.

## I. FUP and FUP2

For dynamic databases, lots of incremental updating techniques have been inveted and developed for mining association rules. One of them previous work for incremental association rule mining is most popular FUP algorithm that was presented by Cheung D W [10]. FUP was the first incremental updating technique to maintain association rules when new data are introduced to the database. The object of FUP is re-using frequent item sets of preceding mining to update with incremental database. The main factor of FUP is pruning technique to impair the number of candidate set in update process. The extension algorithm of FUP is FUP2 [11] which is proposed to maintain with entire update cases when database are added to or deleted from a database.

### II. Negative border

In addition, negative border algorithm [12], incremental mining algorithm based on FUP that reduces to scan original database with updating and keeps track of large-scale itemsets and negative border when transaction is added to database or deleted from database. Negative border consists of all item sets in data set which is candidates of the level wise method. And item sets which are in the negative border do not have enough support but all its subsets are frequent. If the negative border of with large item sets that expands, this algorithm would be required to full scan a entire database. And that is the case because negative border algorithm doesn't cover entire large itemsets in updated database.

### III. *UWEP*

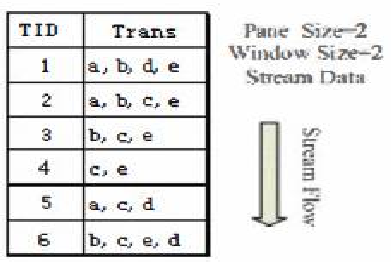
UWEP follows the new approaches of FUP and partition algorithm. It employs a dynamic advanced strategy with updating existing large item sets by detecting , finding and pruning large itemsets consist of original database that won't be longer remain large in updated database. UWEP algorithm scans at most once original and incremental both database. UWEP algorithm [11 that is proposed by D. W. Cheung in in 1997 generates smaller candidate set from the set of itemsets that are large both an original and incremental database. This significantly reduces the number of candidate itemsets in Δ+ it means incremented database. Accordingly, due to these early pruning techniques can step up the efficiency of FUP-based algorithms. The benefit of algorithm UWEP over the FUP-based algorithms is that it prunes the supersets of an originally frequent item set in *D datasets* as soon as it becomes infrequent in the updated database rather than to wait until the $k^{th}$ iteration.

### *IV. NFUP*

To mine new interesting rules with incremented and updated database, NFUP partitions the incremental database according to unit time interval (day, month, quarter or year, for exa.). For every item, suppose that the ending time of exhibition period is analogous. NFUP progressively compiles the phenomenon count of every candidate according to the partitioning characteristics. With-it new information at the last partition of incremental database [13] Therefore, NFUP scans each and every partition backward, that is to say, the last partition is scanned first and the first partition is scanned at last. So NFUP is suited frequently updated databases.

### V. FD-Tree

lun Tan's paper which purpose Fd-tree [5] method which requires no scanning of the whole data set and to only scan the updated transactions or database once without involving candidate sets generation.



| TID | Trans |
|---|---|
| 1 | a, b, d, e |
| 2 | a, b, c, e |
| 3 | b, c, e |
| 4 | c, e |
| 5 | a, c, d |
| 6 | b, c, e, d |

Pane Size=2
Window Size=2
Stream Data
Stream Flow

Fig. 1

In that the method is based on Fd tree method. To evaluate the construction of tree. First, scan the data stream D to get all the occurrences of all the items. Sorting out it in descending order of support. Next, use those occurrences to construct the Fd-tree. That constructed Fd-tree is shown in Figure

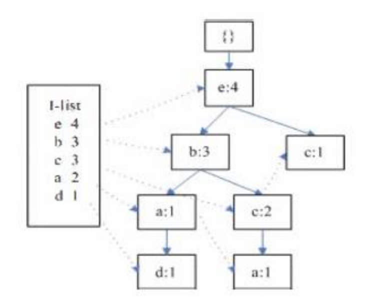


Fig 2. Construction of Fd-tree

Then they introduced how to mine the incremental rules with Fd-tree when old transactions are deleted from data stream D and new transactions are inserted into data stream. When the old pane including T ID 1-2 is deleted. The deletion process never would introduce new branches to the Fd-tree. It will only follow the branch and decreases the count of the nodes. By scanning that deletion part one time, the occurrences of the itemsets are being obtained. The Fd-tree which is deleted after the old pane that is shown in Fig 3.

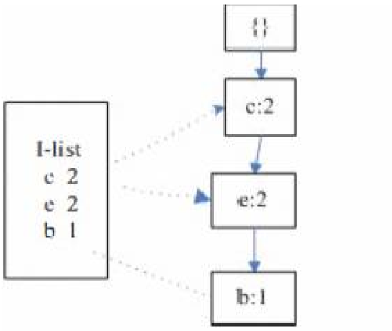


Fig 3.theFd-tree after old pane is deleted

If, the new pane TID 5-6 is inserted. The Fd-tree will only scan the additional transactions only once to update the Fd-tree When the new transactions alloat the same branch with the ones in the original data stream, it just raise the count of each item. Otherwise it will add the new branch into the Fd-tree..

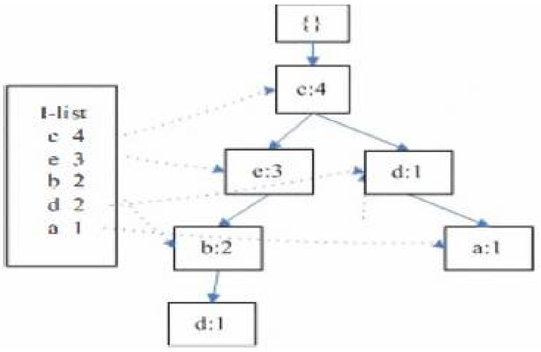


Fig 4.the F d-tree after the new pane is inserted

DOI: 10.15680/IJIRCCE.2016. 0404286

## III. PROPOSED ALGORITHM

In this paper we have proposed new approach for finding frequent itemset& promising frequent item set based on quick sort approach and searching technique without scanning original old database. First original database is scanned and make the table of frequent itemset properly and promising frequent itemset. To find Promising frequent itemset the below formula was found we will use this formula.

$$\min\_\sup_{DB}-\left(\frac{\max supp}{total\ size})\times inc\_size\right)\le \min\_PL<\min\_\sup_{DB}$$

In this paper apriori algorithm is applied to find frequent and promising frequent itemset. It processes with two step prune and join step. In first join step frequent itemset and promising frequent itemset of incremented database can be joined together. For a frequent item, its support count must be greater than it Then, If any k$^{th}$-item has support count greater than or equal to mini_sup(DB∪db), this itemset will be moved to a frequent k-item of an updated database. In the other process, if any k-item has support count less than mini_sup (DB∪db) but it is greater or equal to min_PL(update) , this k-item will be moved to the promise frequent itemset of an updated database. The following formula was developed to update frequent and promising frequent k-items of an updated new database.

$$\min\_PL_{DB\cup db}=\min\_\sup_{DB\cup db}-\left(\frac{\max supp}{total\ size}\times inc\_size\right)$$

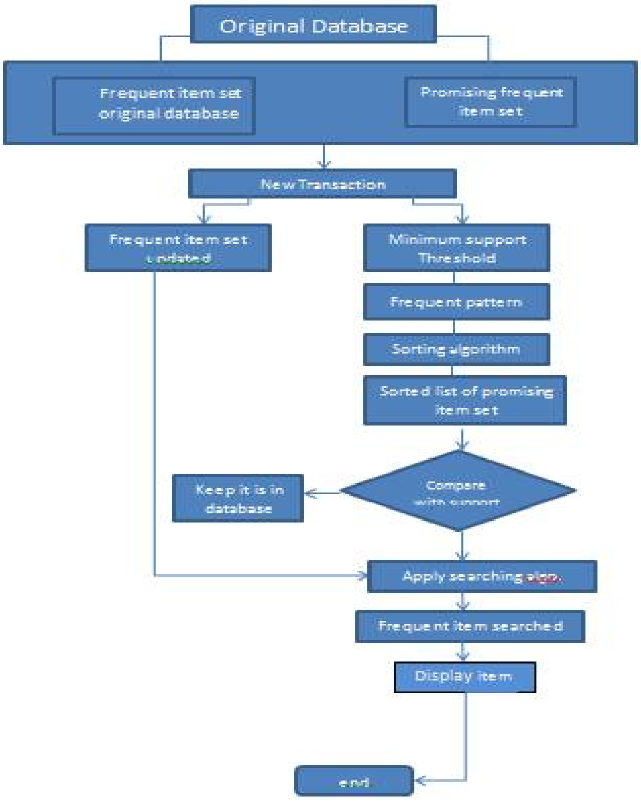


Fig. 5 Proposed System Flow Chart

## IV. PSEUDO CODE

- **Promising data set sorting algorithm.**
  This algorithm for sorting promising data set using quick sort algorithm. First we passed total promising data set to this algorithm also first item set and last item set then find out middle of the item and partition of the total promising data set at the middle of dataset in to two parts and apply this algorithm in each partition of the data set.

```
DBp = Promising Data Set
ISf = First Item Set
ISl = Last Item Set
promisingsort (DBp,ISf, ISl)

1: if ISf ≥ ISl then return
2: q = Partition(DBp,ISf, ISl)
3: Quicksort0 (DBp,ISf, q − 1)
4: Quicksort0 (DBp,q+1, ISl,)

Partitionpromisingsort(DBp,ISf, ISl)
1: x = DBp[ISl]
2: i ← ISf − 1
3: for j ← ISf to ISl − 1 do
4: if DBp[j] ≤ x then
        {
        i ← i + 1
        Exchange DBp[i] and DBp[j]
}
5: Exchange DBp[i + 1] and DBp[ISl]
6: return i + 1
```

**Promising Itemset searching method algorithm.**
This algorithm using for find outs of the required item set in the whole dataset. First we pass the item set in to this algorithm when item set match with current data set then find out its support and return item set and its support count.

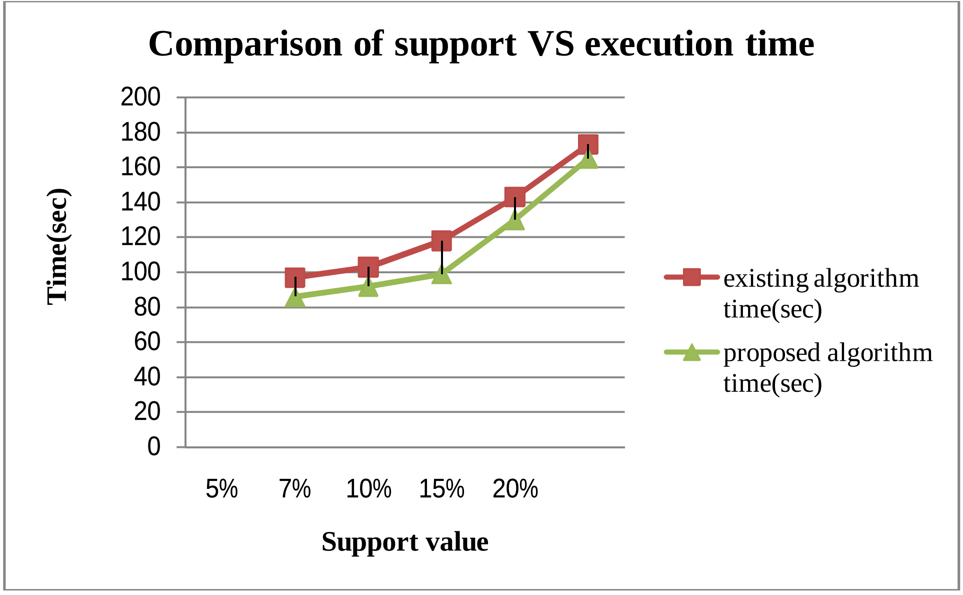
```
DBp = Promising Item set.
SI = Input SerchingItemset
ISp = Output searching Itemset
Support = Output Item set support

Searching(DBp,SI)
    1.  While (DBpi = SI)
        {
        ISp = DBpi
        Support = DBps

        }
    2.  Return(ISp,Support)
```

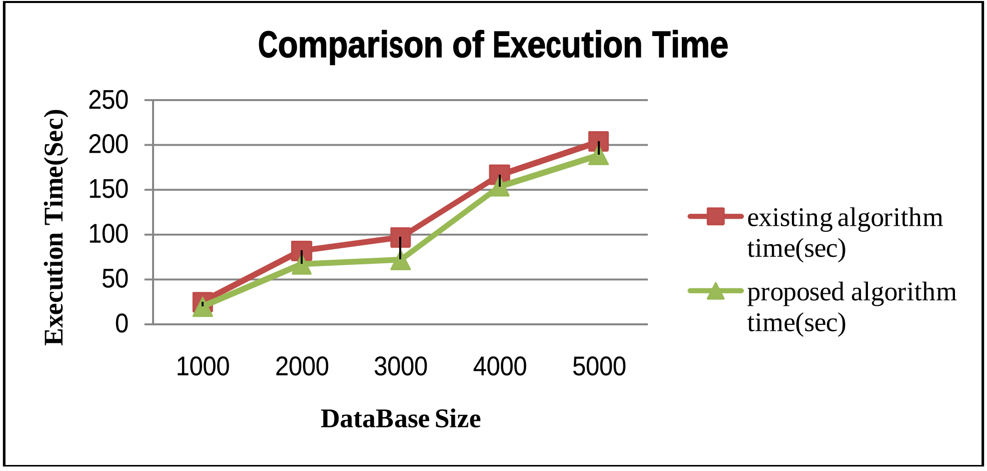
## V. EXPERIMENTAL ANALYSIS

This Graph show comparison of different support values of dataset records and execution time. Support indicates no of item sets upon no of transaction Proposed algorithm have less execution time than existing



TIME COMPARISON WITH DB SIZE

Proposed algorithm has faster execution time than existing.



## VI. CONCLUSION AND FUTURE WORK

This paper determines that the incremental issue brought by the dynamic threshold value for frequent item set and database at the same time, which avoids repeated enumeration. Scanning the original database is very expensive and time consuming in case of dynamic database and has performance bottleneck in the massive data processing so the proposed method avoids the rescanning of the original database when new transactions are added by using promising frequent itemsetby optimizing the algorithm so that it will have faster execution time. The proposed algorithm apply quick sort on the promising itemsets list that will reduce searching time of frequent itemset than existing algorithm and we can mine association information from massive data faster and better. The proposed algorithm can be further enhanced in terms of space complexity. So that algorithm acquire less memory space and get accurate result. Another scope of this algorithm is that currently it is working for incremental dataset in future it can be enhanced to work for decrement dataset as well.

## REFERENCES

1. LijunXu, Yun Zhang A Novel Parallel Algorithm for Frequent Itemset Mining of Incremental Dataset, 2015 IEEE
2. Marghny H. Mohamed ,Mohammed M. Darwieesh, Efficient mining frequent itemsets algorithms, Springer-Verlag Berlin Heidelberg 2013
3. Incremental FP-Growth Mining Strategy for Dynamic Threshold Value and Database Based on MapReduce, Xiaoting Wei, Yunlong Ma , Feng Zhang, Min Liu, WeimingShen, 978-1-4799 3776-9/14/,2014 IEEE
4. R Agrawal, R Srikant, “Fast algorithm for mining association rules,” in Proc. 20th Int. Conf. Very Large Databases (VLDB’94), Santiago, Chile, September 12-15, 1994, pp.487-499.
5. lun Tan, Yingyong Bu and Haiming Zhao, Incremental Maintenance of Association Rules Over Data Streams, 2010 IEEE
6. R. Chang, and Z. Liu, An Improved Apriori Algorithm, International Conference on Electronics and Optoelectronics, 2011, pp. 221-224.
7. Araya Ariya, WorapojKreesuradej, Probability-Based Incremental Association Rule Discovery Using the Normal Approximation, 978-1-4799-1050-2/13, August 2013 IEEE
8. New Incremental Updating Algorithm for Mining Association Rules Based on AprioriTidList Algorithm, Liu Han-bing, Zhang Ya-juan, ZhengQuan-Iu, Ye Mao-gong, July 26-30 2011 IEEE
9. A Novel Parallel Algorithm for Frequent Itemset Mining of Incremental Dataset ,LijunXu ,Yun Zhang, 2015 IEEE
10. Cheung D W, Han J, Ng V T and Wong C Y., “Maintenance of Discovered Association Rules in Large Databases: An Incremental Updating Technique”, *12th International Conference on Data Engineering*, New Orleans, Louisiana, 1996.
11. D. W. Cheung, S.D. Lee, B. Kao, “A General incremental technique for maintaining discovered association rules” , In Proceedings of the 5 th Intl. Conf. on Database Systems for Advanced Applications (DASFAA'97), Melbourne, Australia, April 1997, pp. 185-194.
12. R. Feldman, Y. Aumann, and O. Lipshtat, “Borders: An efficient algorithm for association generation in dynamic databases”,Journal,Intelligent Information System, 1990, pp. 61-73.
13. C.C. Chang, Y.C. Li and J.S. Lee, “An efficient algorithm for incremental mining of association rules”, Proceedings of the 15th international workshop on research issues in data engineering: stream data mining and applications (RIDE-SDMA’05), IEEE, 2005.
14. RatchadapornAmornchewin, Probability-based Incremental Association Rules
15. Discovery Algorithm with Hashing Technique, International Journal of Machine Learning and Computing, Vol.1, No. 1, April 2011
16. JhanviV.Kothari, Probability-based Incremental Association Rules Algorithm Using Hashing Technique, IJARCSMS,Volume 3, Issue 2, February 2015
17. Effective and efficient rule mining technique for incremental dataset, kavithaj.k, 2 manjula d, 3 kasthuribhaj.k,jatit, november 2013. Vol. 57 no.3
18. An Improved Multiple Minimum Support Based Approach to Mine Rare Association Rules, R. Udaykiran, P Krishna Reddy in*2009 IEEE Symposium on Computational Intelligence and Data Mining (IEEE CIDM 2009)* /Report IIIT/TR/2009/24