MINING ASSOCIATION RULES IN INCREMENTAL MINING USING INC_PNAR

[1]G.Srilatha, [2]Dr. N.Subhash Chamdra, [3]A.Vishwanath

[1]Assistant Professor, [2]Professor, [3]Assistant Professor
[1,2,3]Computer Science and Engineering,
[1]Jothishmathi Institute of Technology and Science, Karimnagar,
[2]CVR College of Engineering, Vastu Nagar, Mangalpalli(V), Ibrahimpatnam (M), RangaReddy (D)
[1]Sree Chiatanya College of Engineering, Karimnagar,

Abstract **In many applications the progress of association rules has made very practical in the vein of market basket analysis, decision making, and other varied fields. As the records are continuously added in to the transactional databases and also outdated transactions are useless and are removed, new essential applications need incremental mining. Incremental mining deals with the discovery of association rules acquired from mining of earlier stored databases and incremented databases without scanning the earlier mined databases. Based on this, mining positive and negative association rules is an important research topic. In this paper, proposed an algorithm (INC_PNAR) based on a tree based approach(INC_FIFIT) as a data structure to hold frequent and infrequent itemests, which updates the INC_FIFIT when new transactions are inserted and mine both positive and negative association rules. With a Yule's Coefficient measure, the INC_PNAR algorithm generates all valid positive and negative association rules.**

IndexTerms: **Data Mining, Frequent Itemsets, Infrequent Itemsets, Positive Association Rules, Negative Association Rules**

I. INTRODUCTION

As the time advances, the importance of data mining has grown speedily with the increase of large data in dynamic data bases for various applications. Transactional data bases provide the information related to the behavior of customers and help in making the decisions for improving the quality of business. As the data being processed is huge, it is vital to analyze adequate data appropriately before making decisions. It is essential to develop efficient algorithms to mine association rules from these data. Many applications such as super market data, stock market data, sales data, weather / traffic records, etc., have produced the need for incremental mining, because of the increased use of database-based records where data is added indefinitely. In many applications, recent data is extracted in transaction databases Many applications like super market data, stock market data, sales data, and weather/traffic records, etc, have produced the need of incremental mining, because of increasing the use of record-based databases where data is being endlessly added. In several applications, recent data is mined in the transactional databases. Incremental mining, not only incorporate new data but also eliminate the previous obsolete data from the mining process. Incremental mining deals with the discovery of association rules acquired from mining of earlier stored databases and incremented databases without scanning the earlier mined databases. The endeavor of incremental mining methods is to re-run the mining algorithm on only updated database. The process of incremental mining is shown in Fig 1.

Usually the updated portion is small compared to the whole dataset and earlier mining rules are not used for generating new rules due to their low efficiency. It is essential to develop algorithms in such way that only updated and previous mined rules to be taken into account for generating new rules. Several traditional Association Rule Mining algorithms such as Apriori, AIS, DHS, and Partition were developed for discovering association rules but most of them are static in nature. The first incremental mining algorithm was the Fast–UPdated algorithm (FUP), Several maintenance algorithms were developed such as, FUP, FUP2, UWEP, etc but they were concentrated on only on the problem of maintaining updating positive association rules. Few researchers resolute the importance of negative associations. So, it becomes a challenging issue for maintaining both positive and negative association rules.

II. BASIC CONCEPTS

Let I = $\{i_1, i_2,..., i_n\}$ be a set of n distinct literals called items. Make D a set of transactions, where each transaction T is a set of elements, each transaction is associated with a unique identifier called TID. Let A, called a group of items, be a group of items in I. The number of items in an item group is the length (or size) of an item set. The k-length material groups are referred to as item groups. It is said that transaction T contains A if it is $A \subset T$. The Association rule is implied by model A => B, where $A \subset I$, $B \subset I$, and $A \cap B = \Phi$. The rule A => B contains support (referred to as supp) s in the DB if% s of the transactions in D contain A => B. In other words, support for the rule is the probability that A and B will consolidate together all submitted cases . For example supp (A => B) = supp (AUB) = P (AUB). The rule A=>B has a measure of its strength called confidence (denoted as conf) c if c% of transactions in DB that contain A also contain B In other words, the rule trust is the conditional probability in which B is correct in case of case A for example conf (A => B) = P (B | A) = supp (AUB) / supp (A). The problem of discovering all pairing rules from a set of transactions is to create rules that have greater support and confidence than the given limits. These rules are called strong rules, and the framework is known as the Trust Support Framework for mining the base assemblies.

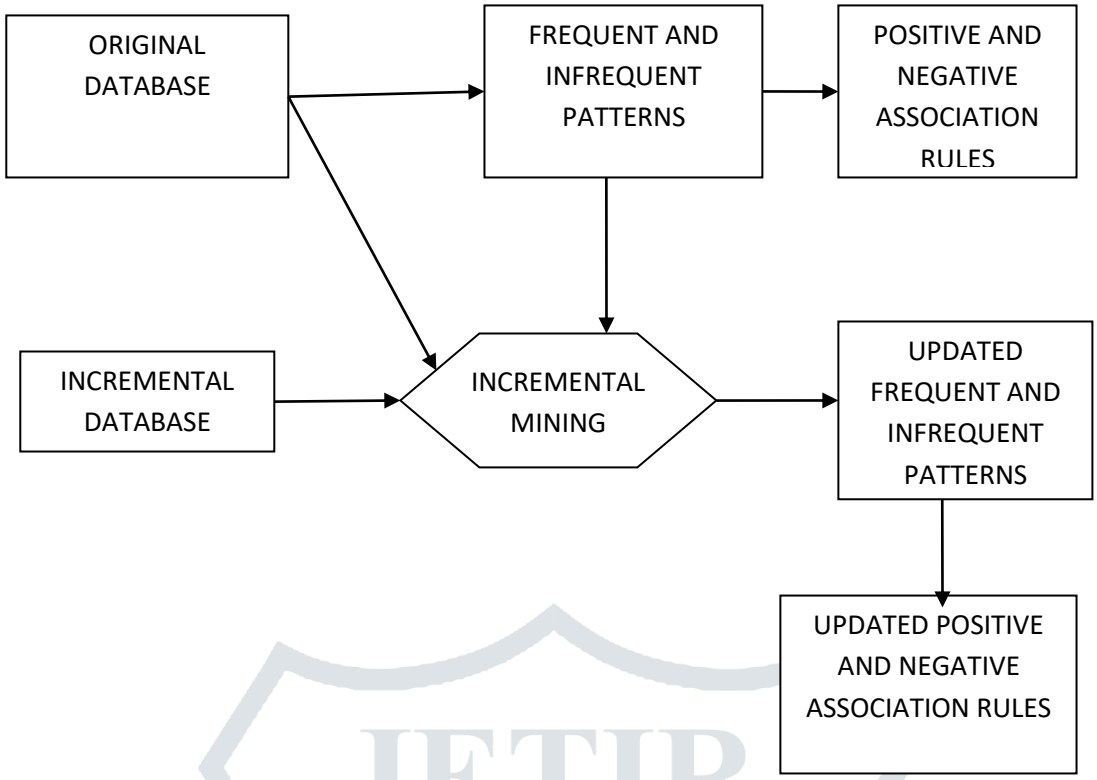

Fig 1: Incremental Mining Process

A **negative association rule** is an implication of the form X =>┐ Y (or ┐ X=>Y or ┐ X=>Y), where X ⊂I, Y ⊂I and X ∩ Y = *Φ,* although the rule in the form of ┐ X =>┐ Y contains negative elements, it is equivalent to a positive association rule in the form of Y=>X. In contrast to positive rules, the negative rule involves the relationship between the frequencies of one set of elements and the absence of the other group of elements. Rule X => has Y has a% s support in the data sets, if% s of the transactions in T contain X elements and no elements of Y. The support for the passive rule base, supp (X => ┐ Y), is the frequency of the occurrence of transactions with element X in the absence of the item Y. Let U is the set of transactions containing all elements in X. The rule X => holds Y in the data set (Database) specified with confidence c%, if c% of transactions in U do not contain the set of elements Y. Confidence of negative association rule, conf (X=> ┐ Y), can be calculated with P(X ┐ Y)/P(X), where P(.) is the probability function. The support and confidence of itemsets are calculated during iterations. However, it is difficult to count the support and confidence of non-existing items in transactions. To avoid counting them directly, we can compute the measures through those of positive rules. The support and confidence of the negative association rules can make use of positive association rules. The support is given by the formulas as,

$$supp(\neg X) = 1 - supp(X)$$
$$supp(\neg X \cup Y) = supp(X) - supp(X \cup Y)$$
$$supp(X \cup \neg Y) = supp(Y) - supp(X \cup Y)$$
$$supp(\neg X \cup \neg Y) = supp(X) - supp(Y) + supp(X \cup Y)$$
$$conf(\neg X => Y) = \frac{supp(X) - supp(X \cup Y)}{1 - supp(X)}$$
$$conf(X => \neg Y) = \frac{supp(X) - supp(X \cup Y)}{supp(X)}$$
$$conf(\neg X => \neg Y) = \frac{1 - supp(X) - supp(Y) + supp(X \cup Y)}{1 - supp(X)}$$

.

III. RELATED WORK

In this section, some related work of researchers as specified.

FUP [1], is the first algorithm to mine the association rules in incremental data mining. It is deprnding on level-wise Apriori algorithm and adopts the pruning strategy used in DHP (Direct Hashing and Pruning) algorithm. First it finds the frequent itemsets from original database and it make use of their support count against the increment database to update with the frequent itemsets of incremental database to find the complete set of frequent itemsets in updated database. FUP requires generating a huge number of candidate sets and needing to scan the original database multiple times. It deals with databases with transaction insertion only but not able to solve the same with transaction deletion. FUP2 [2] is an extension of FUP, which updates the existing association rules when transactions are added to and deleted from the database. FUP2 is similar to FUP for the case of insertion, and is opposite to FUP for the case of deletion.

An algorithm [3] by using negative border, which reduces the I/O requirements for updating the set of large itemsets. This algorithm is based on FUP, reduces to scan original database and keep track of large itemsets and negative border when transaction is added to (or) delete from database. Negative border contains itemsets which are candidates of level-wise method. This approach maintains both frequent and border itemsets. The border item set is not frequent itemsets but all its proper subsets are frequent itemsets. If negative border of large itemsets expands, this algorithm is required to full scan a whole database.

UWEP [4] update with early pruning, follows the approach of FUP and partition algorithm. A dynamic look strategy is used to update existing large item collections by discovering and trimming a large set of elements in an original database that will not remain large in the updated database UWEP scans at most once in original database and exactly once in incremental database. It generates smaller candidate set from the set of itemsets that are large both an original and incremental database and remove those that are no longer remain large after the contribution of the new set of transactions.

Sliding window filtering SWF [5] is a partition algorithm for an incremental association mining. It partitions the database into several partitions and applies the minimum support threshold in each partition to generate candidate itemsets and to filter unnecessary candidate 2-itemsets. The cumulative information is extracted to extract the previous partitions is selectively towards the creation of groups of candidate elements for the subsequent partitions. Requires one scan on the updated database to find the frequent items from candidates.

In [6], adopts an idea of the negative border and the SWF algorithm. Negative border can reduce the number of scans over the original database and SWF algorithm is to discover new itemsets in the updated database. By integrating the SWF with negative border a lot of effort in the re-computation of negative border can be saved and the minimal candidate set of large itemsets and negative border in the updated database can be obtained efficiently.

A new algorithm [7], called Incremental Dynamic Item set counting algorithm, uses the dynamic counting technique to handle the problem of updating association rules.

Pre-large [8] itemsets is introduced and designed a novel, efficient, incremental mining algorithm based on it. This algorithm uses the two user specified upper and low support threshold to define the pre-large itemsets. These pre-large itemsets act as a gap that reduce the movement of itemsets directly from large to small and vice versa in the updated database when transaction are inserted. The proposed algorithm does not require rescanning of the original database until a number of new transactions determined from the two support threshold and the size of the database have been processed. As the database grows the algorithm becomes increasingly efficient.

New Fast Update algorithm (NFUP) [9] introduced a backward method, requires scanning the incremental database rather than rescaning the original database for some of the recurring item sets created in the growing database, aggregating the times when the newly created recurring item sets occur, and deleting the recurring item sets.It does not require to rescan the original database and to discover newly generated frequent itemsets. The proposed method uses information available from a following partition to avoid the rescanning of the original database.

Promising frequent item set algorithm [10] estimate the infrequent itemsets, called promising itemsets by using maximum support count of 1-itemsets obtained from previous mining of an original database. They are capable of being frequent itemsets when new transactions are inserted into the original database. The algorithm can reduce a number of times to scan the original database. False positive item set algorithm [11], uses maximum support count of 1-itemsets obtained from previous mining to estimate infrequent item set false positive item set of an original database. When insert new transactions into an original database, the set of false positive items will be able to be a recurring item set. This paper also suggests a new update and trimming algorithm that ensures that each set of recurring elements from an updated database is found efficiently.

A new algorithm FIM_AIUA [12], which updates association rules by changing the minimum support. This algorithm expands FIM algorithm and AIUA algorithm. It modifies FIM algorithm with a new argument and present a new function fim_aiua_gen () that rewrites the function aiua_gen () of AIUA algorithm. The proposed algorithm requires only one scan for the original database.

In [13], introduced incremental updating algorithm to mine indirect association rules to deal with the maintenance of discovered indirect association rules resulted from the change of the minimum support. The idea is to re-utilize the results acquired in process with the old minimum support. The author used the IS measure as a dependence measure between the itemsets A and B. in this algorithm there are two join steps. One is in first phase to generate the candidate frequent itemsets using Apriori. The other join step is for generating indirect association's C_{k+1} from L_k. Both the two join steps are quite expensive. This strategy requires more space requirement. This algorithm cannot deal with changeable number of transaction in transaction database.

Several algorithms are proposed for mining association rules but an algorithm INAR [14] for mining negative association rules in incremental updating database has proposed. When new database is added to original database, some frequent negative itemsets of original database still become frequent itemsets of original and incremental databases. Some frequent negative itemsets of original database still become infrequent itemsets of original and incremental databases. Some infrequent itemsets of original database become frequent negative itemsets of original and incremental databases. The pruning strategy will not consider the part negative association rules of the form ┐ A =>┐ B and reduce the search space and improve the mining rules and have used the correlation coefficient to judge which form association rules should be mined.

A Frequent and Infrequent itemset tree(FII-tree) as a data structure, a tree based approach has proposed for storing frequent and infrequent itemsets, from which both positive and negative association rules are generated. Initially they have considered frequent-1-itemsets for generating frequent-k-itemsets and infrequent-k-itemsets but not considered the infrequent-1-itemsets. Their method is not applicable for incremental mining an item based vectors BV and finds frequent-1-itemsets and inserts frequent-1-items one by one in the tree and assign to an frequent index. Generate the candidate k-itemsets from frequent (k-1)-itemsets and finds their support count by performing bitwise AND operation between bit vectors (BV). Assigns frequent-k-itemsets to frequent k-itemset list (FL_K) and infrequent-k-itemsets to infrequent k-itemset list (IFL_K) and assigns to frequent index and infrequent index. Reads each frequent k-itemsets from FII tree and generate positive and negative association rules based on threshold values.

IV. PROPOSED METHOD

As the records are continuously added in to the transactional databases and also outdated transactions are useless and are removed, new essential applications have need of the incremental mining. Incremental mining deals with the discovery of association rules acquired from mining of earlier stored databases and incremented databases without scanning the earlier mined databases. Based on this, proposed an algorithm (INC_PNAR) based on a tree based approach(INC_FIFIT) as a data structure to hold frequent and infrequent itemests, which updates the INC_FIFIT when new transactions are inserted and mine both positive and negative association rules. With a Yule's Coefficient measure, the INC_PNAR algorithm generates all valid positive and negative association rules.

Definition 1: The bit vector *(BV)* of an item *i* is in form $BV = (b_1,b_2,....b_m) \in [0, 1]$. *If* $i \in T_k$, and then $b_k=1$, otherwise $b_k=0$, where $k=1,2...m$. The size of bit vector is equal to the number of items in I, and the support of an itemset is equal to the number 1s in the bit vector.

Definition 2: Yule's Coefficient of Association: Association among objects can also be measured efficiently using Yule's coefficient. The merit of this measure is not only we can determine the nature of association,(i.e., whether the attributes are positively associated, negatively associated or independent), but also compute the degree or extent to which the two attributes are associated. Suppose A and B represents the presence of two sets attributes and α and β represents the absence of two sets attributes A and B i.e., α = ¬A and β = ¬B. The Yule's coefficient is denoted by the symbol Q, can be obtained by using

$$Q = \frac{(AB)(\alpha\beta)-(\alpha B)(A\beta)}{(AB)(\alpha\beta)+(\alpha B)(A\beta)} \quad with \quad -1 \le Q \le 1$$

Where,

$$(AB) = support(AB), (\alpha B) = support(\alpha B)$$

$$(A\beta) = support(A\beta), (\alpha\beta) = support(\alpha\beta)$$

When the value of Q is +1 then there is a perfect positive association between the attributes. When Q is -1 there is a perfect negative association between the attributes and when the value Q is zero the two attributes are independent. The coefficient of association can be used to compare the intensity of association between two sets of attributes. Two sets of attributes A and B are said to be associated if they are not independent.

Algorithm: INC_PNAR

Input: db, ms, mc, INC_FIFIT of DB, BV of C_1 candidates of DB
Output: PAR, NAR, FI, IFI
Step 1: Scan the db once, find item based vectors BV of C_{1db}
Find the support count for C_{1db} based on definition 1.
Find F_{1db} , IF_{1db}
Step 2: Consider the cases, update the INC_FIFIT as
Case 1: Check whether F_{1db} is in frindex list (FL_k) of INC_FIFIT; if present, update its support count
- Check the support count of $F_{1(DB+db)} \geq ms_{(DB+db)}$, then no change in INC_FIFIT.

Case 2: Check whether the F_{1db} is in ifrinddex list (IFL_k) of INC_FIFIT; if present update its support count.
- Check the support count of $F_{1(DB+db)} \geq ms_{(DB+db)}$, then update the frindex list (FL_k) and ifrindex list of INC_FIFIT.
- Otherwise, no change in ifrindex list (IFL_k) of INC_FIFIT.

Case 3: Check whether the IF_{1db} is in frindex list (FL_k); if present update its support count.
- Check the support count $IF_{1(DB+db)} \geq ms_{(DB+db)}$, then update the frindex list (FL_k) and ifrindex list (IFL_k) of INC_FIFIT.
- Otherwise, no change in ifrindex list (IFL_k) of INC_FIFIT.

Step 3: Generate candidate k-itemsets C_{kdb} (k=2,3,…..) from F_{1db}
Step 4: Perform the bitwise AND (^) operation between each pair of frequent 1-itemset and find their count.
Step 5: Update INC_FIFIT by following the cases in step2
Step 6: If C_{kdb} is not in INC_FIFIT, then get its item based bit vector BV and find the support count by performing bitwise AND operation between bit vectors.
- Check the support count $supp(C_{k(DB+db)}) \geq ms_{(DB+db)}$, then add the itemset to the frindex list (FL_k) of INC_FIFIT.
- Otherwise add the itemset to the ifrindex list (IFL_k) of INC_FIFIT.

Step 7: Repeat from step 3 to step 6 until all C_{kdb} from F_{1db} are inserted in INC_FIFIT.
Step 8: Generate PAR from $frindex_k$ (FL_k) k=2,3….
For each A, B (A, B = i) and A ∩ B=ϕ, find YuleCoeft(A,B)
If Q>1 then
If conf(A, B) ≥ mc then
Generate positive rules of type A→B,
PAR←PAR U {A→B}
Step 9: Generate NAR from $ifrindex_k$ (IFL_k) k=2,3,….
For each A, B (A, B = i) and A ∩ B=ϕ, find YuleCoeft(A,B)

If Q<1 then
If supp(A→ ┐ B) ≥ $ms_{(D B+ db)}$ and conf(A→ ┐ B) ≥ mc then
Generate negative rules of type A→ ┐ B
NAR←NAR U {A→┐ B}
If Q<1 then
If supp(┐ A→B) ≥ $ms_{(DB + db)}$ and conf(┐ A→B) ≥ mc then
Generate negative rules of type ┐ A→B
NAR←NAR U {┐ A→B}
Step 10: AR ← PAR U NAR

V. EXPERIMENTAL RESULTS

We tested proposed method on a synthetic dataset to study the performance of the algorithm. We provided the solution to sample example. An original database with 10 transactions is shown in Table 1 and incremental transactions with 6 transactions are shown in table 2. Minimum support count is set to 50%. First scan the original database and find the frequent and infrequent itemsets and the tree is shown in fig 2. This tree is taken as input to the proposed algorithm INC_PNAR. We scan the incremental database and updated the frequent and infrequent itemsets which is shown in fig 3. The results are shown in table 3.

Table 1 - Original Database (DB)

TID	Set of Items
1	A B C D E
2	A B C
3	A B D
4	B C D
5	C D E
6	A B C
7	A B C
8	B C E
9	B C D
10	C D

Table 2 - Incremental Database (db)

TID	Set of Items
1	A B C E
2	B C E
3	A B C E
4	C D E
5	A C E
6	A B C E

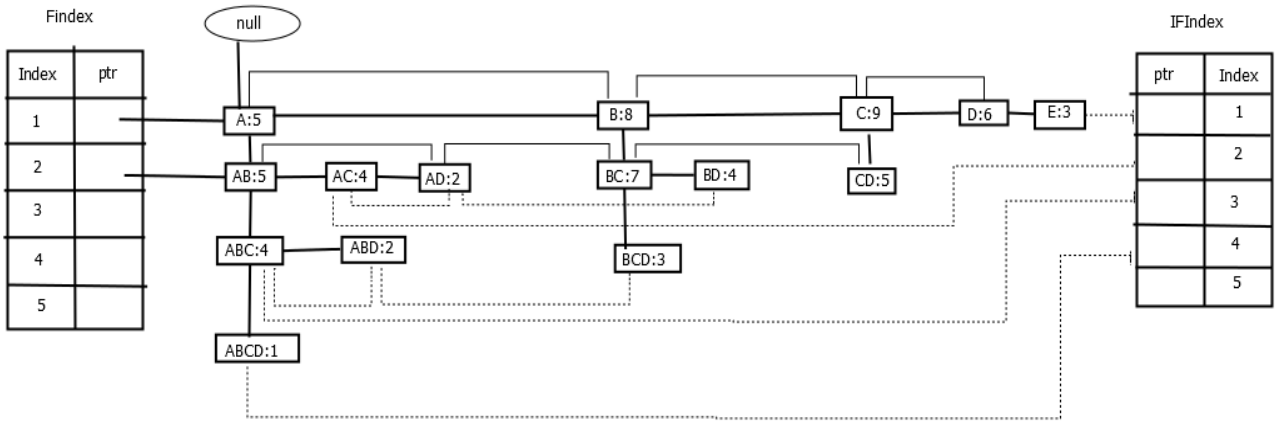

Fig 2: The INC_FIFIT for original database

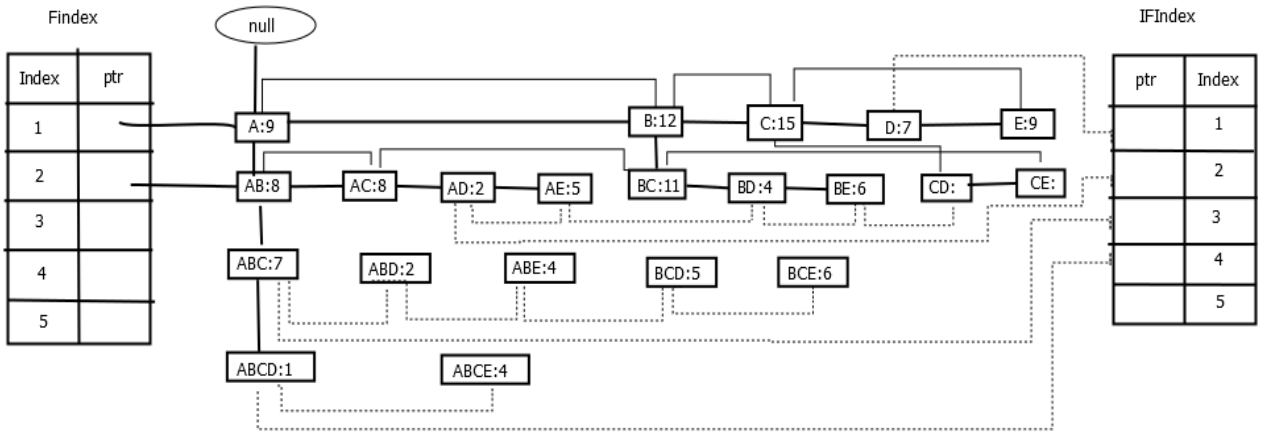

Fig 2: The Updated INC_FIFIT

Table 3 - Results

Supp%	Conf%	Freq Items	Infreq Items	Positive rules	Negative rules	Supp%
10	15	22	2	25	69	10
20	25	17	8	16	51	20
30	35	14	11	10	38	30
40	45	10	11	3	9	40
50	55	8	13	2	5	50
60	65	3	11	0	0	60
70	75	3	8	0	0	70
80	85	1	6	0	0	80
90	95	1	5	0	0	90
16	100	0	6	0	0	16

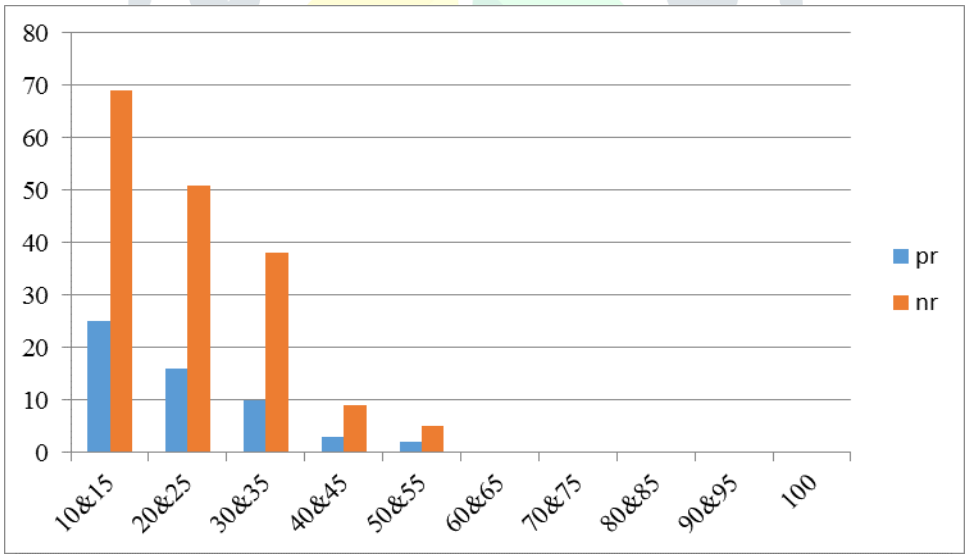

Fig 3: the rules with different minimum support and minimum confidence.

Table 4: Rules with fixed minsupp and different minconf

		no. of rules							
		minsupp & different minconf							
		20 & 25	20 & 35	20 & 45	20 & 55	20 & 65	20 & 75	20 & 85	20 & 95
no. of rules	PR	16	12	8	5	4	0	0	0
	NR	51	42	29	18	7	5	5	5

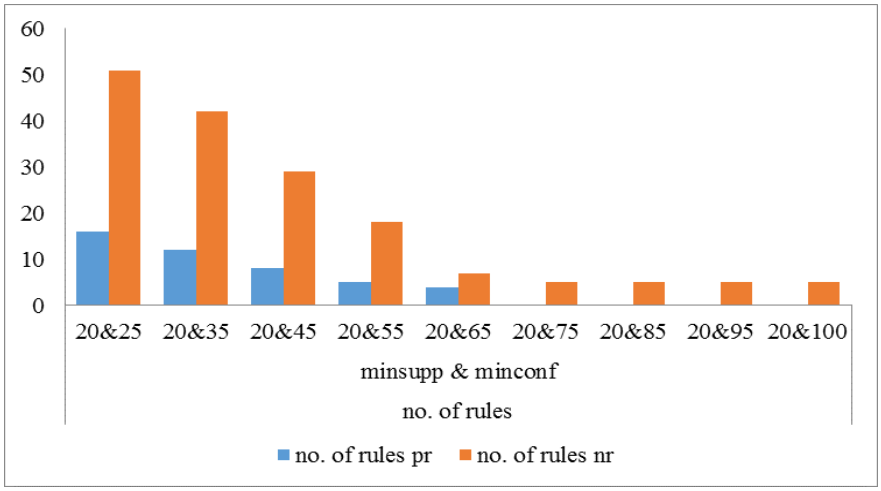

Fig 4: With minimum support and different minimum confidence:

Table 5: Rules with fixed minsupp and different minconf

		No. of Rules				
		Different minsupp & minconf				
		20 & 60	30 & 60	40 & 60	50 & 60	60 & 60
No. of Rules	PR	4	3	1	1	0
	NR	7	7	2	2	0

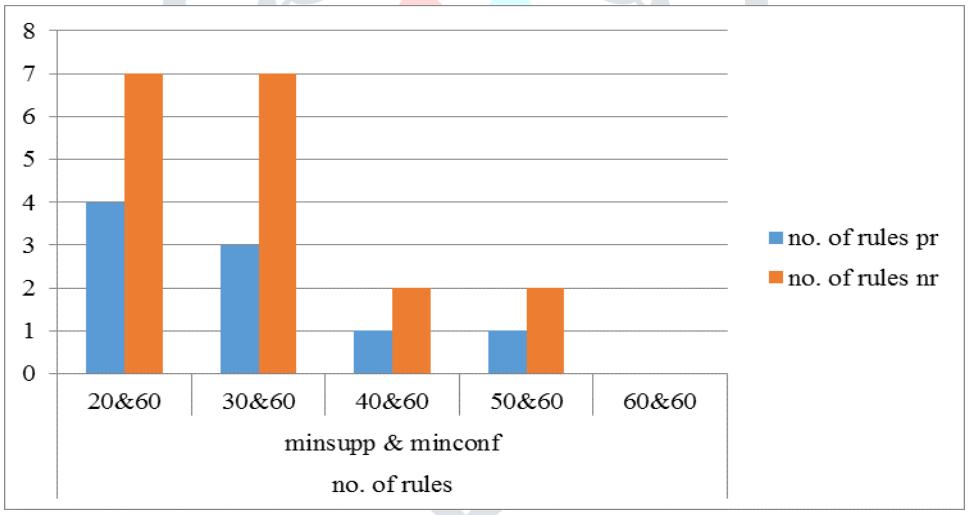

Fig 5: With different minimum support and minimum confidence:

VI. PROPOSED METHOD

According to various works on mining association rules, proposed an algorithm (INC_PNAR) based on a tree based approach (INC_FIFIT) as a data structure to hold frequent and infrequent itemests, which updates the INC_FIFIT when new transactions are inserted and mine both positive and negative association rules. With a Yule's Coefficient measure, the INC_PNAR algorithm generates all valid positive and negative association rules. We conducted experiment on sample dataset. In future we conduct experiments on real datasets and compare the performance of our algorithm with other related algorithms.

REFERENCES

[1] Cheung, David W., et al. "Maintenance of discovered association rules in large databases: An incremental updating technique." Data Engineering, 1996. Proceedings of the Twelfth International Conference on. IEEE, 1996.

[2] Cheung, David Wai-Lok, Sau Dan Lee, and Ben Kao. "A General Incremental Technique for Maintaining Discovered Association Rules." DASFAA. Vol. 6. 1997.

[3] Thomas, Shiby, and et al. "An Efficient Algorithm for the Incremental Updation of Association Rules in Large Databases." KDD. 1997.

[4] Ayan, Necip Fazil, Abdullah Uz Tansel, and Erol Arkun. "An efficient algorithm to update large itemsets with early pruning." Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1999.

[5] Lee, Chang-Hung, Cheng-Ru Lin, and Ming-Syan Chen. "Sliding-window filtering: an efficient algorithm for incremental mining." Proceedings of the tenth international conference on Information and knowledge management. ACM, 2001.

[6] Lee, Wan-Jui, and ShieJue Lee. "A general mining method for incremental Updation in large databases." Systems, Man and Cybernetics, 2003. IEEE International Conference on. Vol. 2. IEEE, 2003. [7] Ng, King-Kwok, and Wai Lam. "Updating of association rules dynamically."Database Applications in Non-Traditional Environments, 1999. (DANTE'99) Proceedings. 1999 International Symposium on. IEEE, 1999.

[7] Hong, Tzung-Pei, Ching-Yao Wang, and Yu-Hui Tao. "A new incremental data mining algorithm using pre-large itemsets." Intelligent Data Analysis 5.2 (2001): 109-128.

[8] Chang, Chin-Chen, Yu-Chiang Li, and Jung-San Lee. "An efficient algorithm for incremental mining of association rules." Research Issues in Data Engineering: Stream Data Mining and Applications, 2005. RIDE-SDMA 2005. 15th International Workshop on. IEEE, 2005.

[9] Amornchewin, Ratchadaporn, and Worapoj Kreesuradej. "Incremental association rule mining using promising frequent item set algorithm."Information, Communications & Signal Processing, 2007 6th International Conference on. IEEE, 2007.

[10] Sun, Li, et al. "An efficient algorithm for updating association rules with incremental transactions and minimum support changes simultaneously."Intelligent Systems (GCIS), 2012 Third Global Congress on. IEEE, 2012.

[11] Amornchewin, Ratchadaporn, and Worapoj Kreesuradej. "False Positive Item set Algorithm for Incremental Association Rule Discovery." International Journal of Multimedia and Ubiquitous Engineering 4.2 (2009).

[12] Zheng, Cheng. "An incremental updating technique for mining indirect association rules." Machine Learning and Cybernetics, 2008 International Conference on. Vol. 1. IEEE, 2008.

[13] Zhu, Honglei, and Zhigang Xu. "A Novel Incremental Updating Algorithm for Maintaining Discovered Negative Association Rules." Research Challenges in Computer Science, 2009. ICRCCS'09. International Conference on. IEEE, 2009.