Journal of Theoretical and Applied Information Technology
30th November 2013. Vol. 57 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

EFFECTIVE AND EFFICIENT RULE MINING TECHNIQUE FOR INCREMENTAL DATASET

[1] KAVITHA J.K, [2] MANJULA D, [3] KASTHURI BHA J.K

[1] Research Scholar, Department of Computer Science and Engineering, Anna University, Chennai, India

[2]Associate Professor, Department of Computer Science and Engineering, Anna University, Chennai, India

[3]Assistant Professor, Department of Electronics and Communication Engineering, SRM University, India

E-mail: [1]–jeyakumar.kavitha@gmail.com , [2] manju@annauniv.edu , [3] kasthuriroshan@gmail.com

ABSTRACT

This paper presents an effective and efficient rule mining technique for incremental dataset. One of the most important challenge is discovering the frequent patterns , if the dataset is incremental in nature , this may cause some existing rules becomes invalid and some new rules becomes valid. In this paper, the Efficient Incremental Rule Mining (EIRM) Algorithm is proposed as a solution to this problem. In the proposed algorithm only a single scan to the database is needed. Instead of itemsets, the Transaction Identifiers (TIDs) are stored for discovering promising and unpromising item sets and relevant support count are also maintained. This helps to find all frequent patterns of an updated dataset efficiently and reduces execution time. The simulation results show that the proposed algorithm has good performance.

Keywords: *Frequent patterns, Association Rules, Data mining, Incremental mining*

1. INTRODUCTION

Data mining (the analysis step of the knowledge discovery in databases process or KDD), a relatively young and interdisciplinary field of computer science is the process of discovering new patterns from large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics and database systems. In data mining, association rule learning is a popular and well researched method for discovering interesting relations between variables in large databases. Association rule plays an important role in data mining and has been becoming applicable in many areas. Since the pioneering works of Apriori [1], AprioriTID [2], partition algorithm [16], FP-tree [9], it has led to many proposals of mining of association rules such as fast mining approaches, updating approaches, and various formations of rule patterns. Association rules are usually required to satisfy a user-specified minimum support and a user-specified minimum confidence at the same time. Association rule generation is usually split up into two separate steps: First, minimum support is applied to find all frequent itemsets in a database. Second, these frequent itemsets and the minimum confidence constraint are used to form rules. While the second step is straight forward, the first step needs more attention. The rules discovered from a dataset only reflect the current state of the dataset. However, in an incremental dataset where the additional new transactions are inserted into the original dataset, keeping patterns up-to-date and discovering new pattern are challenging problems of great practical importance. Because of these update activities, new association rules become valid and some existing association rule would become invalid.

For the updated rules over the total dataset, if the association mining technique redo the rule generation process for the whole dataset, based on the frequent itemsets, simply by discarding the earlier computed results, it will be inefficient. It is generally due to the multiple scanning over the older dataset. If the results of the older dataset are reused for updating the frequent itemsets, then some execution time may be saved. Some of the existing methodologies which attempt to find out the frequent itemsets with minimum number of scanning over the old dataset FUP [4], FUP2 [5], MAAP [7], Borders [8], Modified borders [6]. [10], [12],[15],[13],[17] are some other works , that has given some attention to the incremental rule mining problem.

In this paper, we propose a new rule mining technique for incremental dataset, called Efficient Incremental rule Mining (EIRM) algorithm. The most important challenge is discovering the frequent patterns, if the dataset is incremental in nature. Due to this, some of the existing rules become invalid and some additional new rules

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

become valid. The main goal of our approach is to solve the updating problem of association rules after a number of new records have been added to a dataset. In our approach, we use the Transaction Identifiers (TIDs) are stored as transitional generations instead of itemset, which helps to reduce the frequency of database scans. The algorithm also takes itemset count at each stage into consideration and split the frequent itemset as promising itemset and infrequent itemset as unpromising itemset. Our algorithm requires only a single scan to the dataset. The experimental results show that the running time of the proposed approach is significantly less than that of some previous methods.

2. RELATED WORK

Association mining over incremental dataset is a challenging area of research for the data mining researchers. Several incremental updating techniques have been developed for mining association rules. One of the previous works for incremental association rule mining is FUP [4] algorithm. FUP first scans the incremental part of the dataset and detects (*i*) the looser single itemsets, i.e. the itemsets that become infrequent due to the inclusion of the incremented part and (*ii*) it finds the candidate frequent itemsets. Then the whole dataset (i.e. the old and new together) is scanned to find their support in the complete dataset. Next, it performs similar operations iteratively for *k*-itemsets. Finally, after multiple scanning of the dataset it finds all the maximal frequent sets. As a result, FUP algorithm requires to scan passes over an original database several times when new frequent itemsets are found. This can degrade the performance of FUP algorithm. As an improvement FUP2 [5] algorithm has been introduced to work on a dynamic dataset where new records may be inserted and some of the existing records may be deleted. It extracts the rules from the final dataset by considering both the deleted parts and the newly added part.

To deal with the rescanning problem, Borders [8] approach is presented to find the frequent itemsets from the dynamic dataset, using the frequent itemsets already discovered from the old dataset. Here, an infrequent itemset is termed as border set if all the non empty proper subsets of it are frequent. Due to the insertion of new records to the dataset, some of the border sets may become frequent, and is termed as promoted border set. For that- the border sets of the old dataset also have to be maintained along with the frequent sets derived. Based on the promoted border set, some new candidate itemsets are generated and checked for frequent set. The candidates are generated if there is at least one promoted border set. This algorithm may require more than one pass of the old dataset depending on the frequent sets discovered due to the incremented part. Modified borders [6] algorithm is a modified version of the borders algorithm that minimizes unnecessary candidate generations. However, this algorithm uses an additional user parameter apart from the parameter support count which is sensitive. With proper tuning of these parameters only- a better performance of the algorithm is possible. When this additional parameter's value is closer to the support count, the algorithm converges to the borders algorithm. Although a large number of itemsets in the border itemsets is not become frequent items when a new database is added to an original database, the border-based algorithms still need to keep them in order to guarantee that all frequent itemsets can be found. Thus, the border-based algorithms need large memory space to keep the border itemsets.

To reduce memory space, Tsai et al. [18], Hong et al. [11] and Amornchewin and Kreesuradej [3] propose a new approach. The approach maintains both frequent itemsets and expected frequent itemsets. Based on FUP algorithm, Tsai et al. use not only support threshold but also the degree, called tolerance degree, to finding the expected frequent itemset. For awhile, Hong et al. use the 2 thresholds, upper support and lower support threshold, for mining frequent and expected frequent itmeset. Amornchewin and Kreesuradej proposes the idea for estimate expected frequent itemsets by using maximum support count of 1-itemsets obtained from prior mining. An expected frequent itemset is not a frequent itemset but is expected to become a frequent itemset when a new database is added to an original database. The expected frequent itemsets have lesser members than the border itemsets. As a result the approach uses lesser memory space than the border based approach. In order to guarantee that all frequent itemsets can be found when a new database is added to an original database. But the approaches, can only allow very small size of an increment database to insert into an original database. Therefore, in this paper, an Efficient Incremental Rule Mining (EIRM) algorithm is proposed to speed-up the process of frequent – pattern mining. By storing the Transaction Identifiers (TIDs) of itemsets and precisely calculating support count effectively helps to reduce the required scan iterations to a database. Next section presents our

Journal of Theoretical and Applied Information Technology
30th November 2013. Vol. 57 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

approach which attempts to address the issues efficiently.

3. THE PROPOSED EIRM ALGORITHM

When an incremental dataset is inserted into the original dataset, not only some existing association rules may be invalidated but also some new association rules may be discovered. But due to the addition of these new records, scenario of the rules in the updated dataset may be changed. Some of the new itemsets may become frequent, while some previously derived frequent set may become infrequent. Therefore, an association rule discovery algorithm for a incremental dataset has to maintain frequent itemsets whenever new transactions are inserted. Since all the previous works scans a dataset many times to verify whether the candidate patterns are frequent or not, we may also improve the performance by reducing database scan.

To avoid the problem of multiple scans and to improve performance, the EIRM (Efficient Incremental Rule Mining Algorithm) is proposed in this paper, so the dataset need to be scanned only once. In the proposed algorithm, each transaction has their unique Transaction identifier (TID).By using the hash function concept, to store TIDs in a table structure, helps to calculate the number of itemsets quickly without the need of re-scanning the dataset. Fig. 1 depicts the flow diagram of EIRM (Efficient Incremental Rule Mining).

The algorithm works as 2 subsections. In our approach, an original dataset is firstly mined and all promising and unpromising itemsets are found. Secondly, the incremental dataset in mined and updated to promising and unpromising itemsets. As the result of updation, some unpromising itemsets or new itemsets may be changed into promsing itemset.

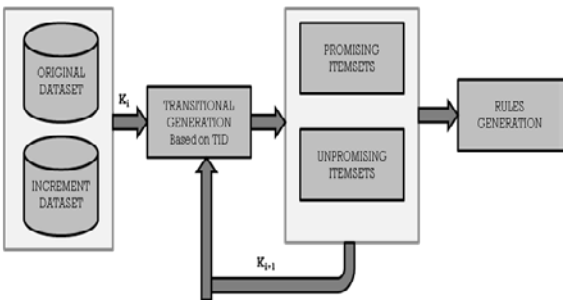


Figure 1: Flow Diagram Of EIRM

3.1 The EIRM Algorithm

The algorithm finds all possible k itemset of promising and unpromising itemsets in original dataset. If member of frequent for each iteration is more than or equal to k-itemset. This idea is to guarantee that EIRM algorithm covers all promising itemset that occur in updated database. Thus, updating the new transactions is quickly because it can use the information from the existing original dataset. Table 1 depicts the IRM Algorithm.

Table 1: The EIRM Algorithm.

The Efficient Incremental Rule Mining Algorithm
Input : A Original dataset DB and an Incremental Dataset db , where each dataset $\{T_0,T_1,T_2\ldots,T_{n-1}\}$in which each transaction T_i $\{i_0,i_1,i_2\ldots.,i_{m-1}\}$; a given minimum support s.
Output: All frequent patterns based on Updated dataset UP.
STEP 1: Scanning the original dataset DB and create the set of transaction identifiers (TIDs) for each item as transitional generation.
STEP 2: Calculate the counts of the transitional 1 - itemsets and sets the itemsets as promising itemset if their counts are greater than the minimum support s otherwise as unpromising itemset.
STEP 3: Set k=1, Where k is the number of itemsets currently processed.
STEP 4: Generate the possible transitional (k+1) – itemsets according to the own k – itemsets and sort the final set.
STEP 5: If further promising itemset is not produced then exit the algorithm, else set k=k+1and repeat the steps 4-5
STEP 6: For the increment dataset db find the set of transaction identifiers (TIDs) for each item and corresponding support count is maintained.
STEP 7: Update all the intermediate transitional generations according to the increment dataset at each iteration.
STEP 8: Finally list the updated promising itemsets of updated dataset UP for rule generation.

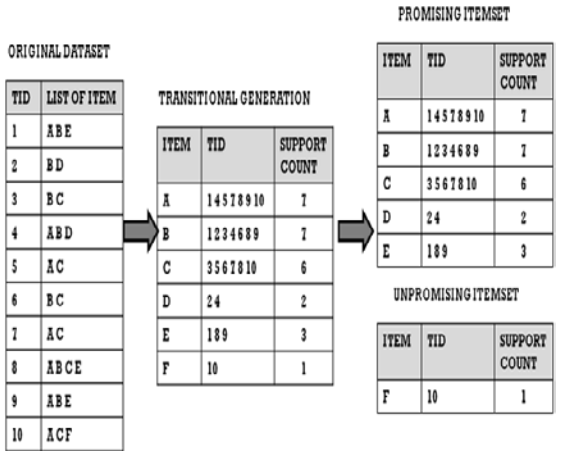
4. AN EXAMPLE

An example given below to illustrate the proposed EIRM Algorithm. An original dataset shown in Fig. 2 has 10 transactions, i.e. |DB|=10.Then, five new transactions is inserted into the original dataset, i.e.|db|=5.Here minimum support count for mining association rules is set to

Journal of Theoretical and Applied Information Technology
30th November 2013. Vol. 57 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. In Step 1, first read and scan the original dataset to build the transitional 1 – itemsets and their TIDs. In step 2, the promising and unpromising itemsets are found based on the support count. The results are shown at the right of Fig. 2.



TID	LIST OF ITEM
1	ABE
2	BD
3	BC
4	ABD
5	AC
6	BC
7	AC
8	ABCE
9	ABE
10	ACF

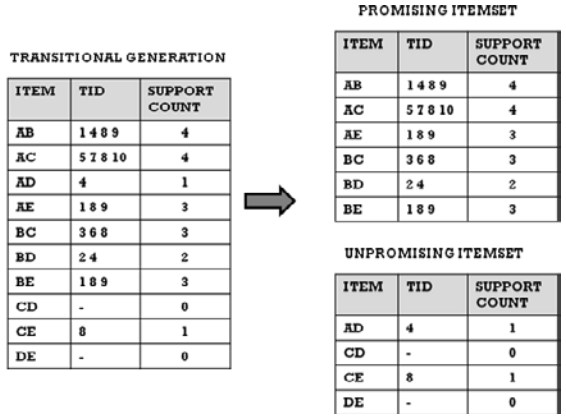
ITEM	TID	SUPPORT COUNT
A	14578910	7
B	1234689	7
C	3567810	6
D	24	2
E	189	3
F	10	1

ITEM	TID	SUPPORT COUNT
A	14578910	7
B	1234689	7
C	3567810	6
D	24	2
E	189	3

ITEM	TID	SUPPORT COUNT
F	10	1

Figure 2: Scanning The Original Dataset To Build Transitional – 1 Itemsets And Their Promising And Unpromising Itemsets.

In step 3 k is initially set at 1, meaning 1-itemsets are first prosessed. In step 4, Generate the possible transitional (k+1) – itemsets according to the own k – itemsets and sort the final set. The results are shown in Fig. 3. In step 5, since the promising – 2 itemsets are not empty, steps 4-5 are then repeated. In this example, only one 3 – itemset is promising. Its related data are shown in Fig. 4. Because only one promising 3- itemset is derived, no 4- itemsets will be formed. The mining process thus ends here for the original dataset.

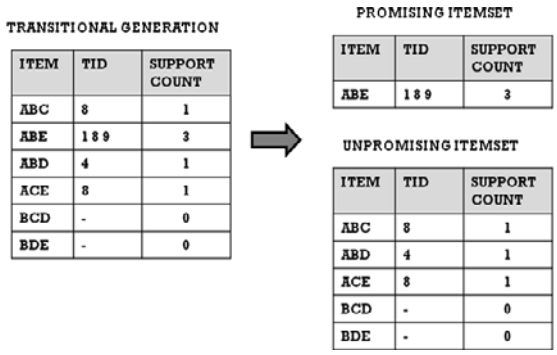


ITEM	TID	SUPPORT COUNT
AB	1489	4
AC	57810	4
AD	4	1
AE	189	3
BC	368	3
BD	24	2
BE	189	3
CD	-	0
CE	8	1
DE	-	0

ITEM	TID	SUPPORT COUNT
AB	1489	4
AC	57810	4
AE	189	3
BC	368	3
BD	24	2
BE	189	3

ITEM	TID	SUPPORT COUNT
AD	4	1
CD	-	0
CE	8	1
DE	-	0

Figure 3: Transitional – 2 Generations And Corresponding Promising And Unpromising Itemsets



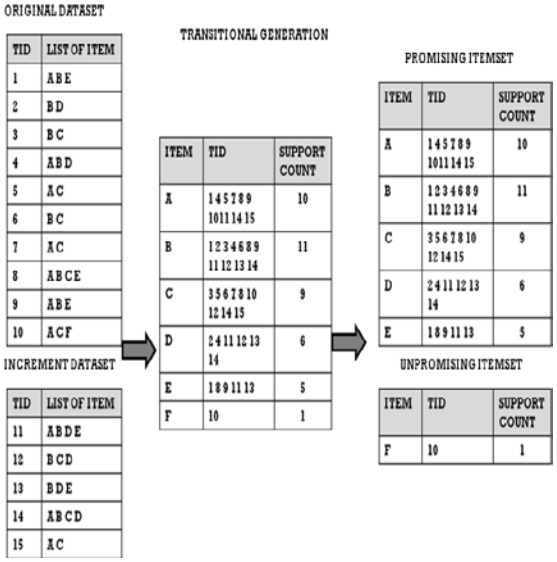
ITEM	TID	SUPPORT COUNT
ABC	8	1
ABE	189	3
ABD	4	1
ACE	8	1
BCD	-	0
BDE	-	0

ITEM	TID	SUPPORT COUNT
ABE	189	3

ITEM	TID	SUPPORT COUNT
ABC	8	1
ABD	4	1
ACE	8	1
BCD	-	0
BDE	-	0

Figure 4: Transitional – 3 Generations And Corresponding Promising And Unpromising Itemsets

In step 6, read and scan the incremental dataset db to find the set of transaction identifiers (TIDs) for each item and corresponding support count is maintained. In step 7, update all the intermediate transitional generations according to the increment dataset at each iteration.



TID	LIST OF ITEM
1	ABE
2	BD
3	BC
4	ABD
5	AC
6	BC
7	AC
8	ABCE
9	ABE
10	ACF

ITEM	TID	SUPPORT COUNT
A	1457891011 1415	10
B	1234689 11121314	11
C	3567810 121415	9
D	2411121314	6
E	1891113	5
F	10	1

ITEM	TID	SUPPORT COUNT
A	1457891011 1415	10
B	1234689 11121314	11
C	3567810 121415	9
D	2411121314	6
E	1891113	5

TID	LIST OF ITEM
11	ABDE
12	BCD
13	BDE
14	ABCD
15	AC

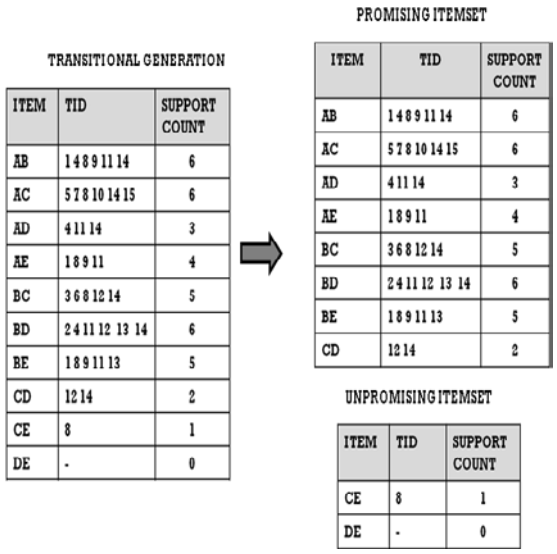
ITEM	TID	SUPPORT COUNT
F	10	1

Figure 5: Updation Of The Transitional – 1 Itemsets Based On Incremental Dataset

The results are shown in Fig. 5 to Fig.7. Since, it is easier to update the already stored information using hash function, the execution time of the incremental dataset requires only less time. The promising itemsets in the original dataset DB are {AB, AC, AE, BC, BD, BE, ABE} where as in the updated dataset UP {AB, AC, AD, AE, BC, BD, BE, CD, ABC, ABD, ABE, BCD}. In which {AD, CD, ABC, ABD, BCD} are promising itemsets after the inclusion of incremental dataset.

Journal of Theoretical and Applied Information Technology
30th November 2013. Vol. 57 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

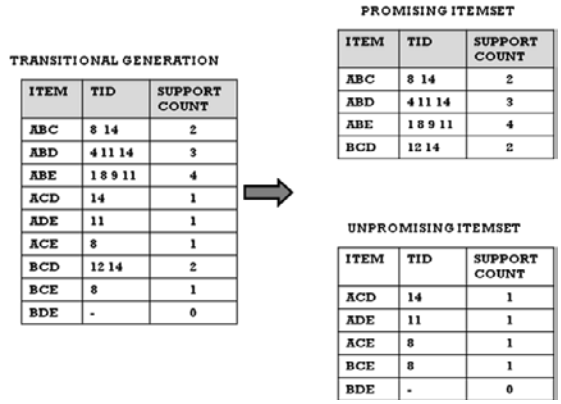


ITEM	TID	SUPPORT COUNT
AB	1 4 8 9 11 14	6
AC	5 7 8 10 14 15	6
AD	4 11 14	3
AE	1 8 9 11	4
BC	3 6 8 12 14	5
BD	2 4 11 12 13 14	6
BE	1 8 9 11 13	5
CD	12 14	2
CE	8	1
DE	-	0

ITEM	TID	SUPPORT COUNT
AB	1 4 8 9 11 14	6
AC	5 7 8 10 14 15	6
AD	4 11 14	3
AE	1 8 9 11	4
BC	3 6 8 12 14	5
BD	2 4 11 12 13 14	6
BE	1 8 9 11 13	5
CD	12 14	2

ITEM	TID	SUPPORT COUNT
CE	8	1
DE	-	0

Figure 6: Updation Of The Transitional – 2 Itemsets Based On Incremental Dataset.



ITEM	TID	SUPPORT COUNT
ABC	8 14	2
ABD	4 11 14	3
ABE	1 8 9 11	4
ACD	14	1
ADE	11	1
ACE	8	1
BCD	12 14	2
BCE	8	1
BDE	-	0

ITEM	TID	SUPPORT COUNT
ABC	8 14	2
ABD	4 11 14	3
ABE	1 8 9 11	4
BCD	12 14	2

ITEM	TID	SUPPORT COUNT
ACD	14	1
ADE	11	1
ACE	8	1
BCE	8	1
BDE	-	0

Figure 7: Updation Of The Transitional – 3 Itemsets Based On Incremental Dataset

.

5. EXPERIMENTAL RESULTS

The main purpose of this work is to find out an alternative way to improve the performance of the incremental association rule mining. To evaluate the performance of Efficient Incremental Rule Mining (EIRM) algorithm, the algorithm is tested on a PC with a 2.8 Ghz Pentium 4 Processor, and 1 GB main memory. The experiments are on a synthetic dataset, called T10I4D10K. The technique for generating the dataset is proposed by Agrawal [2]. The synthetic dataset comprises 1,00,000 transactions has 10 items on average and the maximal size itemset is 4.

The proposed algorithm is used to find association rules from an original database of 20,000 transactions. We show that the performance of our algorithm with minimum support 4%. The same sizes of increment databases, i.e. 10 % of the original database, are added to the original database for 100 trails. For comparison purpose, FUP, Borders and pre-large algorithm are also used to find association rules from the same original database and the same increment database. Table 2 and Fig. 8 show the average execution time of FUP, Borders, pre-large and the proposed algorithm. The results show that the proposed algorithm has much better running time than that of the other algorithms.

Table 2: The Average Execution Time Of 100 Trails For FUP, Borders, Pre-large And EIRM

Minimum support	FUP	Borders	Pre-large	EIRM
4%	5900	7588	4207	1800

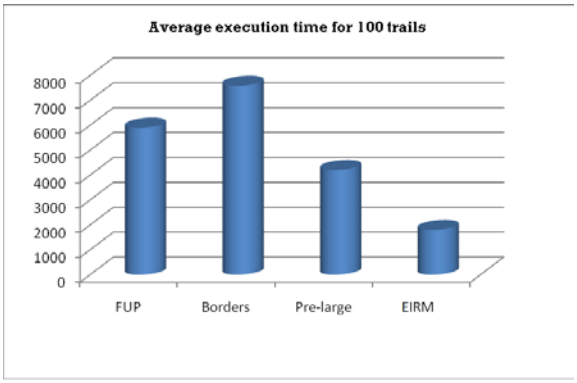


Figure 8: The Average Of Execution Time In FUP, Borders, Pre-large And EIRM With Minimum Support 4%.

6. CONCLUSION

Discovering frequent patterns in an incremental dataset is a sensible research topic. The process of generating itemsets at each level and identifying the frequent itemsets, however, very time consuming. In this paper, the Efficient Incremental Rule Mining (EIRM) algorithm is proposed to solve this problem. It stores the TIDs of items in a table using hash function to compute the occurrences of itemsets fast. EIRM algorithm can thus effectively reduce the required scan iterations to a dataset. It also adopts useful partitions of promising and unpromising itemsets helps to reduce unnecessary transitional generations. Experimental results show that EIRM has better performance than previous works. Thus the proposed algorithm can thus provide a useful strategy for incremental mining problems.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFRENCES:

[1] Agrawal R, Imielinski T and Swami A., "Mining Association Rules between Sets of Items in Large Databases", *ACM SIGMOD Int'l Conference on Management of Data,1993*.

[2] Agrawal R and Srikant R., "Fast Algorithms for Mining Association Rules in Large Databases", *Proceedings of 20th Int'l Conference on VLDB*, August- September, Chile,1994.

[3] Amornchewin, R., and Kreesuradej, W., "Incremental association rule mining using promising frequent itemset algorithm "; *In Proceeding 6th International Conference on Information, Communications and Signal Processing, Singapore*, 2007.

[4] 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.

[5] Cheung D W, Lee D W, and Kao S D ., "A General Incremental Technique for Maintaining Discovered Association Rules", *In Proceedings of the Fifth International Conference on Database System for Advanced Applications*, Melbourn, Australia,1997.

[6] Das A, Bhattacharyya D. K. ., "Rule Mining for Dynamic Databases", *AJIS*, **13**, No.1, pp 19-39,2005.

[7] Ezeife C I and Su Y., "Mining Incremental Association Rules with Generalized FP Tree", *Proceedingsof 15th Canadian Conference on Artificial Intelligence*, AI2002, Calgary, Canada, May,2002.

[8] Feldman R, Aumann Y and Lipshtat O ., "Borders : An Efficient Algorithm for Association Generation in Dynamic Databases", *Journal of Intelligent Information System*, Pages 61–73,1999.

[9] Han J, Pei J and Yin Y., "Mining Frequent Patterns without Candidate Generation", *Proceedings of the 2000 ACM-SIGMOD International Conference on Management of Data*, Dallas, Texas, USA,2000.

[10]Hong T-P., Lin C-W., Wu Y-L., "Incrementally Fast Updated Frequent Pattern Trees", *Expert Systems with Applications: An Int'l Journal*, **34**, no. 4, pp 2424-2435, May,2008.

[11]Hong, T.P., Wang, C.Y., and Tao, Y.H., "A new incremental data mining algorithm using pre-large itemsets"; *Journal of Intelligent Data Analysis*, Vol. 5, No.2 2001, 111-129,2001.

[12] Huang J-P., Chen S-J., Kuo H-C., "An Efficient Incremental Mining Algorithm-QSD", *Intelligent Data Analysis*, **11**, No. 3, pp 265-278, August,2007.

[13] Kao B., Zhang M., Yip C-L., Cheung D. W., Fayyad U., "Efficient Algorithms for Mining and Incremental Update of Maximal Frequent Sequences", *Data Mining and Knowledge Discovery*, **10**, No. 2, pp 87-116, March,2005.

[14]Li J., Manoukian T., Dong G., Ramamohanarao K.., "Incremental Maintenance on the Border of the Space of Emerging Patterns", *Data Mining and Knowledge Discovery*, **9**, No. 1, pp 89-116, July,2004.

[15]Ou J-C., Lee C-H., Chen M-S., "Efficient Algorithms for Incremental Web Log Mining with Dynamic Thresholds", *The International Journal on Very Large Data Bases*, **17**, No. 4, pp 827-845, July,2008.

[16]Savasere A, Omiecinski E and Navathe S., "An Efficient Algorithm For Mining Association Rules in Large Databases", *Proceedings of 21st Conference on Very Large Databases*, Zurich, September,1995.

[17]Tseng M-C., Lin W-Y., Jeng R., "Incremental Maintenance of Generalized Association Rules Under Taxonomy Evolution", *Journal of Information Science*, **34**, No. 2, pp 174-195, April,2008.

[18] Tsai, Paury S.M.,Lee, Chih-Chong, and Chen, Arbee L.P., "An Efficient Approach for Incremental Association Rule Mining"; *Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining, Lecture Notes In Computer Science*,Vol. 1574 archive, 1999.