

Partition-Based Approach for Fast Mining of Transitional Patterns

R.B.V Subramanyam and Soma Raju Suvvari

Department of Computer Science and Engineering

National Institute of Technology

Warangal, India

{rbvs66, soma2402}@gmail.com

Abstract-Studying the dynamic behaviour of patterns whose frequency changes over time is of recent interest. Such patterns are named as transitional patterns [1] and were found only upon mining the frequent patterns in a transaction database. In this paper, a partition-based algorithm is proposed to find a variation of transitional patterns. Our approach do not require a set of frequent patterns mined over the database to initiate the process of finding transitional patterns and thus reduces the number of scans over the database. In [1] the range of T_{ξ} was fixed, whereas we expressed it in terms of support so as it varies with the sizes of the databases and minimum pattern support value.

Keywords:- Data Mining, Frequent patterns, transitional patterns, significant milestones.

I. INTRODUCTION

Frequent Pattern Mining is the crucial step in Association Rule Mining and also in other related Data Mining functionalities. It includes sequential pattern mining, structured pattern mining and correlation mining. Candidate set generation-and-test paradigm of Apriori kind algorithms, FP-tree based algorithms have influenced several researchers to contribute with innovative methods. In parallel, focus was also set on several kinds of frequent patterns including closed [15, 16], maximal [17], non-derivable [18], utility patterns [14] for non-redundancy and compactness of association rules useful for decision making. Considering temporal information, various algorithms proposed for mining temporal patterns including cyclic patterns [6], calendar-based patterns [5]. Time-stamps of the transactions are normally ignored in finding frequent patterns related to Association Rules. Earlier researchers treated all the transactions uniformly in a database. Recently, Wan and An [1] have introduced “Transitional Patterns” which represent patterns whose frequency dramatically changes over time in a transaction database. Primarily they focused on discovering time points at which positive (or negative) transitional patterns increases (or decreases) their frequency significantly over time. Their study gave scope to disclose the dynamic behaviour of the frequent patterns as they considered the time-stamps of transactions in mining process. Applications of transitional patterns include tuning the marketing strategies in retail environments, analyzing the time points at which drugs cause the side effects in diagnosis, restructuring web links based on frequency of visiting the web pages. TP-Mine Algorithm [1] runs in two

stages. At the first stage, a traditional frequent mining algorithm is invoked to mine all frequent patterns and this stage demands at least two scans over the database. In the second stage, an additional scan over database identifies transitional patterns among the discovered frequent patterns along with transitions and significant ascending (descending) milestones in the transaction database.

II. CONTRIBUTIONS

Contributions of this paper are as follows.

(i) Reducing the number of scans over huge databases is an objective in Data Mining processes. Our approach enumerates all the transitional patterns and their significant milestones with only two scans over the database. Unlike in [1], finding the set of all frequent itemsets in advance is not a pre-requisite for our process. Partition-based approach is used for optimal memory utilization. In the first scan, we adopt a strategy that collects a set of candidate itemsets for the second scan along with their p-support values. Out of these candidate itemsets, transitions, significant ascending (descending) milestones of frequent itemsets will be listed at the end of the second scan. Definitions of Transitions, significant milestones are presented in later sections.

(ii) Transitions occur at the very beginning and tail end of the database has been ignored by Wan and An [1] as such transitional patterns are sporadic. Range of such ignoring portions of the databases was fixed by them with the help of T_{ξ} . However, we have defined T_{ξ} in terms of User-defined minimum support value so as ignoring portions vary with sizes of the databases and also with support values. This avoids losing non-sporadic patterns which may not happen in case fixed range is adopted.

III. RELATED WORK

The frequent pattern mining was introduced by Agrawal and srikanth [2], and it has been studied by many researchers. As a result, many algorithms [3, 4] have been proposed for extracting the frequent patterns. The problem of association rule mining has been extended in several ways after its introduction; incremental mining [19], mining of generalized and quantitative rules [13], non-derivable itemset mining [18], utility mining [14], and mining of temporal association rules [15, 16].

Recently researchers are considering the time stamps into the transaction database, so that the dynamic nature of the

mined patterns can be revealed by these methods. The itemsets whose supports increase significantly from one dataset to another are named as emerging patterns [12, 20] and are used to capture the emerging trends in time stamped documents, or useful contrasts between data classes. Bay and Pazzani [11] present the problem of mining contrast-sets, which are conjunctions of attribute-value pairs that have meaningfully different support levels across the contrasting groups. Many previous studies also consider the time stamps in the transaction database when mining the frequent patterns. Such studies include the mining of sequential patterns [7, 8, 9] and mining frequent episodes [10]. Temporal association rule is an association rule that holds in specific time intervals. There are several kinds of temporal association rules, including cyclic association rules (i.e., the association rule that occur periodically over time) [6], calendar based temporal association rules [5], etc. Ann and Wan [1] have introduced the Transitional Patterns that represent patterns whose frequency dramatically changes over time in a transaction database. The transitional pattern captures the dynamic behaviour of the transactional database. They introduce the concept of significant milestones for each transitional pattern, which are specific time points at which the frequency of the pattern increases or decreases most significantly. They presented an algorithm named as TP-Mine algorithm which works in two stages.

IV. PRELIMINARY CONCEPTS

As in Wan and An [1], we assume that the transactions in a transaction database D are ordered by their time-stamps, the position of a transaction t in D , denoted by $\rho(t)$ and the support of itemset X from the i 'th transaction to the j 'th transaction is denoted by $\text{sup}(X, i..j)$. In addition to that the cover of an itemset X in D denoted by $\text{cov}(X, D)$, is the set of transactions in D that supports itemset X . The following definitions in this section are taken from [1].

1. The i 'th milestone of a pattern X in D is denoted by $\xi^i(X)$, is the relative position (expressed in percentage) of the transaction t in D , t is the i 'th transaction in $\text{cov}(X, D)$.
2. The support of a pattern X before (after) its i 'th milestone in D , denoted by $\text{sup}^i(X)$ ($\text{sup}^i_+(X)$), is the ratio of i ($\text{cov}(X) - i$) to the position of transaction t in D , t is the i 'th transaction in $\text{cov}(X, D)$.
3. The transitional ratio of pattern X at its i 'th milestone in D is denoted by $\text{tran}^i(X)$, is the ratio of $(\text{Sup}^i_+(X) - \text{Sup}^i(X))$ to the maximum value among the $\text{Sup}^i_+(X)$ and $\text{Sup}^i(X)$.
4. Transitional Pattern (TP): - A pattern X is a transitional pattern in D if there exists at least one milestone of X , $\xi^k(X)$ belongs to T_ξ , such that: $\text{sup}^k_+(X) \geq t_s$, $\text{sup}^k(X) \geq t_s$ and $|\text{tran}^k(X)| \geq t_t$. Where t_s , t_t are pattern support threshold and transitional pattern threshold. Moreover X is called the positive transitional pattern (PTP) when $\text{tran}^k(X) > 0$ and X is

called the negative transitional pattern (NTP) when $\text{tran}^k(X) < 0$.

5. Significant frequency ascending milestone (SFAM):- A milestone M , which is in the range of T_ξ is the significant frequency ascending milestone for a positive transitional pattern X with respect to the time period T_ξ with $\text{sup}^M(X) \geq t_s$, and if the transitional ratio of X at the milestone M is greater than or equal to the transitional ratios of all other milestones of X which are in the range of T_ξ .
6. Significant frequency descending milestone (SFDm):- A milestone M , which is in the range of T_ξ is the significant frequency descending milestone for a negative transitional pattern X with respect to the time period T_ξ with $\text{sup}^M(X) \geq t_s$, and if the transitional ratio of X at the milestone M is greater than or equal to the transitional ratios of all other milestones of X which are in the range of T_ξ .
7. The freq ($X^{r,n}$) denotes the number of milestones of pattern X from the r 'th partition to the n 'th partition.

V. PROPOSED APPROACH

We proposed a two phase approach for extracting the frequent patterns, positive (negative) transitional patterns and their significant milestones. We scan the database once in the first phase and we extract the candidate itemsets in this phase. The second phase is used to discover the frequent patterns, transitional patterns and their significant milestones by scanning the database once again.

- p-support(x):- The p-support of an itemset X is the support of X computed based on transactions accumulated up to the current partition from its induction.

A. Range of T_ξ

In order to obtain the reliable values for $\text{sup}^i(X)$ and $\text{sup}^i_+(X)$, the number of transactions before $\xi^i(X)$ and after $\xi^i(X)$ should not be too small. In [1] the range of T_ξ was fixed, and as a result we lose the portions of the database which reveals the true transitions of frequent patterns. We have defined T_ξ in terms of user-defined minimum support threshold, it considers all the milestones of a pattern existing in the transactions from $u*s*|D|+1$ transaction to the transaction $|D|-(u*s*|D|)-1$, where u is a user defined parameter and $0 < u < 1$. As a result the ignoring portions of the database vary with sizes of the databases and also with support values. This has reduced the need of expert for setting the range of T_ξ .

B. First Phase

The partition based approach is used to accomplish the task of identifying the candidate itemsets in the database D , where the database D is logically divided into n non-overlapping partitions of $P_1, P_2, P_3 \dots P_n$.

Procedure: Finding the candidate itemsets

Find Candidates (Database D, Support s)

```
{
1. Partition D into n equal partitions of P1, P2, P3 ... Pn.
2. Load the database in to memory, partition-wise.
3. Scan the loaded partition in to the memory.
4. At the end of each partition decision is taken on an
   itemset X in maintained list of candidates C, regarding
   either to keep it in the list or to remove from the list.
   i. An itemset X is inducted into the list if it is newly
      found in the current partition and it gains the
      minimum support in that partition.
   ii. An itemset X that is already in the list is removed, if
      the p-support of the itemset X is less than
      minimum support threshold.
   iii. Itemset X will be kept in the list with it's p-support,
      if it's p-support is greater than or equal to the
      minimum support threshold.
5. Return C.
}
```

At the end of the first scan, list C contains a few itemsets with their *p-support* values and are considered as the potential globally-frequent candidates. The itemsets which carry the support count from the first partition itself are declared them as a frequent itemsets.

Clusters induced by candidate itemset X: - For each candidate itemset X, we can have two clusters of transactions D₁(X) and D₂(X) of D such that X is infrequent in the first cluster (i.e. first r-1 partitions) and X is frequent (i.e. last n-r+1 partitions).

Result-1:- Let D be database and partitioned into n equal size partitions having size |P|. If s is minsup and r is the partition number at which a candidate X is inducted in to the list at the latest, then X is frequent in D, if $(s * |D| - \text{freq}(X^{r,n})) \geq \text{freq}(X^{1,r-1})$.

Proof:- It is obvious that if $(s * |D| - \text{freq}(X^{r,n})) > \text{freq}(X^{1,r-1})$, then the itemset X is infrequent in the database D.

C. Second Phase

Extraction of transitional patterns and their significant milestones in the second cluster is a straight forward one, but it is not the case in the first cluster. In the first cluster, to identify a pattern X as a transitional pattern and a mile stone i as a significant mile stone, we need to know the $\text{sup}_+^i(X)$ of the pattern X. Thanks to the properties of a candidate itemset, where it divides the database into two clusters, and for any frequent itemset X, at any milestone i in the first cluster, the $\text{sup}_+^i(X)$ is always greater than or equal to the minimum support, and is always greater than pattern support threshold. In the first cluster we try to identify the milestones which may become significant for a candidate itemset X. For

every candidate itemset X, at the beginning of the second cluster, we decide whether it is frequent or not and the milestone(s) which were suspected as a significant are really significant or not. Though, if it is not a significant we use this information to find whether it is transitional pattern or not.

D. Significant Frequency-Descending (Ascending) Milestone:-

The milestones i and j (present in the transactions t₁, t₂ with time stamps u and v and t₂ is in the k'th partition) are identified as candidates for the significant frequency-descending (ascending) milestones for a candidate itemset X in the first cluster, such that the time stamp of j is greater than the time stamp of i. According to the properties of candidate itemset the $\text{sup}_+^i(X)$ and $\text{sup}_+^j(X)$ are always greater than the minimum support threshold. The $\text{sup}_+^i(X)$ and $\text{sup}(X, u..v)$ are to be considered for the identification of most probable candidate for the significant frequency-descending (ascending) milestones of a candidate itemset X.

Lemma-1:- The milestone i' lying in D₁(X) is potential significant frequency descending milestone for itemset X if $\text{Sup}_+^i(X) > s$.

Proof: - According to the properties of candidate itemset, any candidate itemset X and at any mile stone i the $\text{sup}_+^i(X)$ is always greater than or equal to the minimum support threshold, otherwise X will become infrequent in the database D. If the $\text{sup}_+^i(X)$ is not greater than minimum support threshold, then X will not become a negative transitional pattern.

We formulated some claims with which one can prune out some milestones from considering them as candidates for the significant milestones. During the second scan, at the end of each partition, all related candidate frequent itemsets are considered for deciding their milestones if any, and sometimes for declaring as globally infrequent. At the end of the second scan the algorithm reports the frequent itemsets, transitional patterns and their significant milestones.

Let i and j (with i < j) be the two milestones of a pattern X with time-stamps u and v.

Claim-1:- The i'th milestone is a significant frequency descending milestone for X if it satisfies either of the following.

1. $\text{Sup}(X, u+1...v) < s$.
2. $\text{Sup}(X, u+1...v) < \text{Sup}_+^i(X)$, $\text{Sup}(X, u+1...v) > s$ and $\text{Sup}(X, u+1...v) \leq (s * |D| - i) / (|D| - i)$.

Claim-2:- The j'th milestone is the significant frequency descending milestone for X, if $\text{Sup}(X, u+1...v) \geq \text{Sup}_+^j(X)$.

Claim-3:- The j'th milestone is the significant frequency ascending milestone for X if it satisfies any one of the condition.

1. $\text{Sup}_-^i(X) \geq s$ and $\text{Sup}_-^j(X) < s$.
2. $\text{Sup}_-^i(X) \geq s$, $\text{Sup}_-^j(X) \geq s$ and $\text{Sup}(X, u+1...v) \leq \text{Sup}_-^i(X)$.

3. $\text{Sup-}i(X) < s$, $\text{Sup-}j(X) < s$ and $\text{Sup}(X, u+1 \dots v) \leq \text{Sup-}i(X)$.

Claim-4:- The i 'th milestone is the significant frequency ascending milestone for X if $\text{Sup}(X, u+1 \dots v) \geq \text{mpsp}$ and it satisfies any one of the condition.

1. $\text{Sup-}i(X) < s$ and $\text{Sup-}j(X) \geq s$.
2. $\text{Sup-}i(X) \geq s$, $\text{Sup-}j(X) \geq s$, $\text{Sup}(X, u+1 \dots v) > \text{Sup-}i(X)$.
3. $\text{Sup-}i(X) < s$, $\text{Sup-}j(X) < s$ and $\text{Sup}(X, u+1 \dots v) > \text{Sup-}i(X)$.

Note:- Here mpsp is the maximum possible $\text{Sup}_+^i(X)$ in D .

In the claim-1 and claim-4 the j 'th milestone and in the claim-2 and claim-3 the i 'th milestone can be ignored from the consideration of potential significant milestones.

Proof: - Here is a sketch of the proof referred to the claims mentioned above. Let w, x, y, z , and q are the $\text{sup}_+^i(X)$, $\text{sup}_+^j(X)$, $\text{sup}_+^i(X)$, $\text{sup}_+^j(X)$ and $\text{sup}(X, u+1 \dots v)$. The milestone $i(j)$ is the significant frequency descending (ascending) milestone if, $xy > wz$ ($wz > xy$), since from the definitions of transitional ratio and significant ascending (descending) milestones. From the claim-1 first condition, we have, $y > w$, since q is infrequent. From the lemma-1, we have, $x > z$. From these two conclusions we can derive $xy > wz$. It proves the Claim-1's first condition. From the claim-3's second condition, we have, $x \geq z$, since $q \leq x$. We have also, $y > w$, because i is a candidate for a significant frequency ascending milestone. From these two conclusions we can derive $xy > wz$. This proves the Claim-3's second condition. The remaining conditions of the claims follow the same procedure for to prove them correct. In the following we are presenting the FastTP-Mine algorithm.

E. Algorithm: Finding the transitional pattern.

FastTP-Mine (Database $P_1, P_2 \dots P_n$,
Support s, t_s, t_p)

```
{
  1. C=Find Candidates (D, s).
  2. for all  $k \leftarrow 1$  to  $n$  {
  3.   for each  $t$  in  $P_k$  {
  4.     for each subset  $e$  of  $t$  and if  $e$  in  $C$  {
  5.       increment the support count of  $e$ ;
  6.       if  $\rho(t)$  in  $T_\xi$  {
  7.         if  $t$  is in the first cluster of  $e$  and is a  $j$ 'th milestone of  $e$  in  $D$  {
  8.           if  $\text{sup}_+^j(e) \geq t_s$  {
  9.             Find-probable-Milestone using the claims;
  10.          }
  11.        } else if it is in the second cluster and it is first milestone of  $e$  in the second cluster {
  12.          Resolve-Candidate-Milestones ( $e$ ) //resolve all of its candidate significant milestones of  $e$ 
  13.        } Else if ( $\text{sup}_+^i(e) \geq t_s$ ) {
  14.          if  $\text{tran}^i(e) \geq +t_i$  {
```

```
15.   Declare  $e$  as a positive transitional pattern if not already declared and test it for significant frequency ascending milestone.
16.   } else if  $\text{tran}^i(e) \leq -t_i$  {
17.   Declare  $e$  as a negative transitional pattern if not already declared and test it for significant frequency descending milestone.
18.   }
19.   }
20. }
21. }
22. }
23. Remove the infrequent itemsets and respective candidate milestones.
24. }
} //end
```

In the above algorithm the step 9 uses the Claims mentioned above for deciding the most probable candidate for SAM or SDM, if they fail to decide then, then this milestone is considered as candidate for both SAM and SDM. The 23'rd step uses the Result-1 for removing the infrequent itemsets.

F. Procedure: Resolve-Candidate-Milestones

Resolve-Candidate-Milestones (itemset e)

```
{
  1. For each candidate milestones  $u$  of  $e$  {
  2.   if  $e$  is frequent {
  3.     If ( $\text{sup}_+^u(e) \geq t_s$ ) {
  4.       If ( $\text{tran}^u(e) \geq +t_i$ ) {
  5.         Declare  $e$  as a positive transitional pattern and store the milestone which gives maximum transitional ratio.
  6.       } else If ( $\text{tran}^u(e) \leq -t_i$ ) {
  7.         Declare  $e$  as a negative transitional pattern and store the milestone which gives minimum transitional ratio.
  8.       }
  9.     }
  10.   }
  11. }
} //end
```

G. Database scans

The FastTP-Mine algorithm runs in only two scans and is the vast improvement over the TP-Mine algorithm in [1].

VI. EXPEREMENTAL RESULTS

We tested our FastTP-Mine algorithm on our own synthesized database and summaries of the results shown in the following tables. The TABLE I show the characteristics of the first synthesized database and its corresponding number of frequent patterns and transitional patterns shown in the TABLE II.

TABLE I. DATABASE-1 CHARACTERISTICS

Database	Number of items	Number of transactions	Partition size
Synthetic Database-1	125	16000	1000

TABLE II. PATTERN GENERATED FROM DATABASE-1

Number of candidate itemsets	Number of frequent patterns	Number of PTPs	Number of NTP
8,654	695	32	61
Minsup=0.5%		$t_s=0.15$	$t_t=0.3$

The TABLE-III shows the characteristics of our second synthesized database and the number of frequent patterns and transitional patterns shown in the TABLE-IV.

TABLE III. DATABASE-2 CHARACTERISTICS

Database	Number of items	Number of transactions	Partition size
Synthetic Database-2	85	24000	4000

TABLE IV. PATTERNS GENERATED FROM DATABAS -2

Number of candidate itemsets	Number of frequent patterns	Number of PTPs	Number of NTP
1986	569	41	26
Minsup=0.6		$t_s=0.15$	$t_t=0.3$

The graph (Figure1) shows how the numbers of candidates are varying with increasing the partition size.

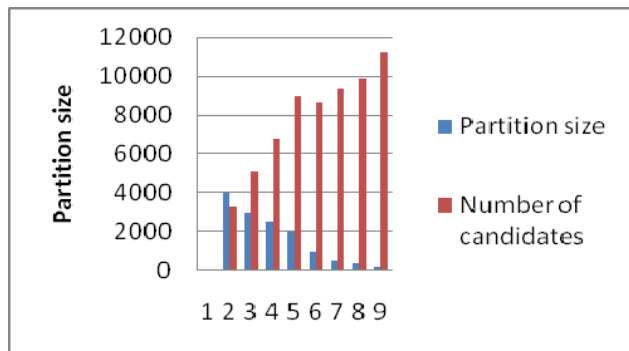


Figure I. Performance comparison between partition size and number of candidate itemsets

VII. CONCLUSIONS

Fast TP-Mine algorithm is proposed to mine transitional patterns in less number of scans of the database compared to TP-Mine algorithm and with less computational time.

REFERENCES

- [1] Q. Wan, A. An, Discovering Transitional Patterns and their significant milestones in a transaction databases - IEEE transactionson Knowledge and Data Engineering, Dec',2009.
- [2] R. Agrawal and R. Srikant, Fast Algorithms for Mining Association Rules, Proc. 20th Int'l Conf. Very Large Data Bases,pp. 487-499, 1994.
- [3] R. Agrawal, T. Imielinski, and A. Swami, Mining Association Rules between Sets of Items in Large Databases, Proc. 1993 ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '93), pp. 207-216, 1993.
- [4] Mohammed. J. Zaki and Karam. Gouda, Fast Vertical Mining Using Diffsets, Ninth ACM SIGKDD'03 Int'l Conf. Knowledge Discovery and Data Mining, pp. 326-335, 2003.
- [5] Y.J. Li, P.Ning, X.Seen Wang, and S.Jajodia, Discovering Calendar-Based Temporal Association Rules, Proc. Eighth Int'l Symp. Temporal Representation and Reasoning (TIME '01), pp. 111-118, 2001.
- [6] B. Ozden, S. Ramaswamy, and A. Silberschatz, Cyclic Association Rules, Proc. 14th Int'l Conf. Data Eng. (ICDE '98), pp. 412-421, 1998.
- [7] Z. X. Liao and W. C. Peng, Mining sequential patterns across multiple sequence databases, Data & Knowledge Engineering 68 (2009) 1014-1033.
- [8] R. Agrawal and R. Srikant. Mining sequential patterns. In proc. 1995 Intl. Conf. Data Engineering. (ICDE'95), pages 3-14, Taipei, Taiwan, March 1995.
- [9] J. Han, J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, and M. C. Hsu, FreeSpan: Frequent pattern-projected sequential pattern mining . In SIGKDD'00, Boston, MA, Aug. 2000.
- [10] H. Mannila, H. Toivonen, and A. Inkeri Verkamo, Discovery of Frequent Episodes in Event Sequences, Data Mining and Knowledge Discovery, vol. 1, pp. 259-289, 1997
- [11] S. D. Bay and M. J. Pazzani, Detecting Change in Categorical Data: Mining Contrast Sets, Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '99), pp. 302-306, 1999.
- [12] G. Z. Dong and J. Y. Li, Efficient Mining of Emerging Patterns: Discovering Trends and Differences, Proc. Fifth ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '99), pp. 43- 52, 1999.
- [13] R. Srikant and R. Agrawal, Mining Generalized Association Rules, 21st VLDB conference , Zurich , Switzerland, 1995.
- [14] C. F. Ahmed, S. K. Tanbeer, B. S. Jeong, and Y. K. Lee, "Efficient Tree Structures for High Utility Pattern Mining in Incremental Databases", IEEE transactionson Knowledge and Data Engineering, vol.21, no.12, December 2009.
- [15] J. Pei, J. Han, and R. Mao, CLOSET: An efficient algorithm for mining frequent closed itemsets. In DMKD'01 workshop, Dallas, TX, May 2001.
- [16] X. Yan, J. Han, and R. Afshar, CloSpan: Mining Closed Sequential Patterns in Large Databases. In SDM'03, San Francisco, CA, May 2003.
- [17] D. Burdick, M. Calimlim, and J. Gehrke. MAFIA: A Maximal Frequent Itemset Algorithm, IEEE transactionson Knowledge and Data Engineering, Vol. 17, No. 11, November 2005.
- [18] T. Calders, B. Goethals, Depth-first non-derivable itemset mining, In Proc. SIAM Int. Conf. on Data Mining SDM'05.
- [19] S. Thomas, S. Bodagala, K. Alsabti, S. Ranka, An efficient algorithm for the incremental updation of association rules in large databases, in: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97), Newport Beach, California, 1997
- [20] J. Bailey, T. Manoukian and K. Ramamohanarao, Fast Algorithms for Mining Emerging Patterns, Sixth European Conference on Principle of Data Mining and Knowledge Discovery (PKDD'02), pp 39-50.