ETP-Mine: An Efficient Method for Mining Transitional Patterns

B. Kiran Kumar[1] and A. Bhaskar[2]

[1]Department of M.C.A., Kakatiya Institute of Technology & Science, A.P. INDIA.
kirankumar.bejjanki@gmail.com
[2]Department of M.C.A., Kakatiya Institute of Technology & Science, A.P. INDIA.
bhaskar_adepu@yahoo.com

ABSTRACT

A Transaction database contains a set of transactions along with items and their associated timestamps. Transitional patterns are the patterns which specify the dynamic behavior of frequent patterns in a transaction database. To discover transitional patterns and their significant milestones, first we have to extract all frequent patterns and their supports using any frequent pattern generation algorithm. These frequent patterns are used in the generation of transitional patterns. The existing algorithm (TP-Mine) generates frequent patterns, some of which cannot be used in generation of transitional patterns. In this paper, we propose a modification to the existing algorithm, which prunes the candidate items to be used in the generation of frequent patterns. This method drastically reduces the number of frequent patterns which are used in discovering transitional patterns. Extensive simulation test is done to evaluate the proposed method.

KEYWORDS

Frequent Patterns, Association Rules, Transitional Patterns, Significant milestones.

1. INTRODUCTION

Frequent pattern mining was extensively studied in the area of data mining research. These frequent patterns are used in many different data mining tasks like Association rule mining [2] [3] [12], sequential pattern mining [4], structured pattern mining [7], correlation mining [5], and associative classification [9]. Many algorithms were proposed to find frequent patterns in transaction databases namely Apriori [3], FP-growth [8], and Eclat [15]. But these methods generate large number of patterns when the minimum support threshold is low. Out of these patterns most are not required in data mining tasks. To avoid these useless or redundant patterns, new types of patterns were introduced, namely maximal frequent itemsets [1] [6] [11], and closed frequent itemsets [10].

All the above methods do not consider time stamps associated with the transactions. So, the above methods do not reveal the dynamic behavior of the patterns. For example, in an electronics shop database, the sales for Air conditioners will be high during the summer season and the sales will be low in other seasons. If we consider total transactions, Air conditioners will become frequent, but these are frequent in only summer season. To capture the dynamic behavior of the patterns, new type patterns were introduced, called Transitional patterns [13] [14]. Transitional patterns can be both positive and negative. Positive transitional patterns increase their frequency

DOI : 10.5121/ijdms.2010.2303

at some time stamp. Negative transitional patterns decrease their frequency at some time stamp. Significant milestone of the transitional pattern is the time point at which the frequency of the transitional pattern changes most frequently. TP-Mine algorithm [14] was proposed to discover the transitional patterns and their significant milestones.

The TP-Mine algorithm contains two major steps. In the first step all the frequent patterns are generated from the transaction database. In the second step these frequent patterns are used in generating transitional patterns and their significant milestones within the given milestone range that the user is interested. But, some of the frequent patterns generated in the first step are not used in the second step.

In this paper we are proposing a modification to the TP-Mine algorithm to eliminate the useless frequent patterns generated in the first step. Thereby we can reduce the number of computations required in generating transitional patterns.

The rest of the paper is organized as follows. Section 2 describes preliminaries and definitions used in TP-Mine algorithm. Section 3 describes TP-Mine Algorithm. In section 4, we describe the modified TP-Mine (ETP-Mine) algorithm. Section 5 presents experimental results of our work. Finally, in section 6 we conclude our work.

2. PRELIMINARIES AND DEFINITIONS

Mining frequent patterns is one of the fundamental operations in data mining applications for extracting interesting patterns from databases. Let I= $\{i_1, i_2 \ldots i_n\}$ be a set of items. Let D be a set of database transactions where each transaction T is a set of items and ||D|| be the number of transactions in D. Given X= $\{i_j \ldots i_k\} \subseteq$ I ($j \leq k$ and $1 \leq j, k \leq n$) is called a pattern. The support of a pattern X in D is the number of transactions in D that contains X. Pattern X will be called frequent if its support is no less than a user specified minimum support threshold.

Definition 2.1: The cover of an itemset X in D, denoted by *cov(X,D)*, is the number of transactions in which the item X appears.

Definition 2.2: An itemset X in a transaction database D has a support, denoted by *sup(X, D)*, which is the ratio between *cov(X, D)* to the number of transactions in D i.e.,||D||.

$$\sup(X, D) = \frac{cov(X,D)}{||D||}$$

Definition 2.3: Assuming that the transactions in a transaction database D are ordered by their time-stamps, the position of a transaction T in D, denoted by ρ(T), is the number of transactions whose time-stamp is less than or equal to that of T. Thus $1 \leq \rho(T) \leq ||D||$.

Definition 2.4: The i^th transaction of a pattern X in D, denoted by $T^i(X)$, is the *i*^th transaction in *cov(X)* with transactions ordered by their positions, where $1 \leq i \leq cov(X, D)$.

Definition 2.5: The i^th milestone of a pattern X in D, denoted by $\xi^i(X)$, is defined as

$$\xi^i(X) = \frac{\rho(Ti(X))}{||D||} \times 100\% \quad \text{where } 1 \leq i \leq cov(X).$$

Definition 2.6: The support of the pattern X before its i^th milestone in D, denoted by $sup_{-}^{i}(X)$ is defined as

$$sup_{-}^{i}(X) = \frac{i}{\rho(\mathrm{Ti(X)})} \quad \text{where } 1 \leq i \leq cov(X).$$

Definition 2.7: The support of the pattern X after its ith milestone in D, denoted by $sup_{+}^{i}(X)$ is defined as

$$sup_{+}^{i}(X) = \frac{cov(x)-i}{||\mathrm{D}||-\rho(\mathrm{Ti(X)})} \quad \text{where } 1 \leq i \leq cov(X).$$

Definition 2.8: The Transitional ratio of the pattern X at its i^{th} milestone in D is defined as

$$tran^{i}(X) = \frac{sup_{+}^{i}(X) - sup_{-}^{i}(X)}{Max(\, sup_{+}^{i}(X), sup_{-}^{i}(X))} \quad \text{where } 1 \leq i \leq cov(X).$$

Definition 2.9: A pattern X is a transitional pattern (TP) in D if there exists at least one milestone of X, $\xi^{k}(X) \in T_{\xi}$, such that:

$$1.\ sup_{-}^{k}(X) \geq t_s \ and\ sup_{+}^{k}(X) \geq t_s\ and \quad 2.\ |tran^{i}(X)| \geq t_{t,}$$

where T_{ξ} is a range of $\xi^{i}(X)$ ($1 \leq i \leq cov(X)$), t_s and t_t are called pattern support threshold and transitional pattern threshold, respectively. X is called a Positive Transitional Pattern (PTP) when $tran^{k}(X)>0$; and X is called a Negative Transitional Pattern (NTP) when $tran^{k}(X)<0$.

Definition 2.10: The significant frequency-ascending milestone of a positive transitional pattern X with respect to a time period T_{ξ} is defined as a tuple, $(\xi^{M}(X), tran^{M}(X))$, where $\xi^{M}(X) \in T_{\xi}$ is the Mth milestone of X such that

$$1.\ \ sup_{-}^{M}(X) \geq t_s; \ \text{ and } \ 2.\ \forall \xi^{i}(X) \in T_{\xi}, tran^{M}(X) \geq tran^{i}(X)$$

Definition 2.11: The significant frequency-descending milestone of a negative transitional pattern X with respect to a time period T_{ξ} is defined as a tuple, $(\xi^{N}(X), tran^{N}(X))$, where $\xi^{N}(X) \in T_{\xi}$ is the Nth milestone of X such that

$$1. sup_{+}^{N}(X) \geq t_s;\ and\ 2. \forall \xi^{i}(X) \in T_{\xi}, tran^{N}(X) \leq tran^{i}(X)$$

3.TP-MINE ALGORITHM

Consider the following transaction database shown in Table 1.

Table 1: Example Database

TID	List of Item IDs	Time stamp
001	P1,P2,P3,P5	Nov, 2005
002	P1,P2	Dec, 2005
003	P1,P2,P3,P8	Jan, 2006
004	P1,P2,P5	Feb, 2006
005	P1,P2,P4	Mar, 2006
006	P1,P2,P4,P5,P6	Apr, 2006
007	P1,P2,P3,P4,P6	May, 2006
008	P1,P4,P6	Jun, 2006
009	P4,P5,P6	Jul, 2006
010	P1,P2,P3,P4,P5,P6	Aug, 2006
011	P1,P3,P4,P6	Sep, 2006
012	P1,P3,P5	Oct, 2006
013	P1,P2,P3,P6,P7	Nov, 2006
014	P1,P3,P4,P5	Dec, 2006
015	P1,P3,P4	Jan, 2007
016	P1,P2,P3,P5	Feb, 2007

TP-Mine algorithm [14] generates the set of positive and negative transitional patterns with their significant milestones.

Input: A transaction database (D), milestone range (T_ξ), pattern support threshold(t_s), and transitional pattern(t_t).
We considered D as set of transactions listed in Table 1, T_ξ= {25%, 75%}, t_s=0.05, t_t=0.5

This algorithm has two major phases. In the first phase all frequent itemsets along with their support counts are generated using Apriori [3] or FP-growth [8] with t_s as minimum support threshold. In this step, the algorithm generates n number of frequent patterns whose support ≥ t_s from the transaction database D. The resultant frequent patterns (n=87) are shown in Table 2.

Table 2: Set of frequent patterns

FP	Sup	FP	Sup	FP	Sup	FP	Sup
P1	15	P3,P5	5	P1,P6,P7	1	P1,P2,P5,P6	2
P2	10	P3,P6	4	P2,P3,P4	2	P1,P2,P6,P7	1
P3	10	P3,P7	1	P2,P3,P5	3	P1,P3,P4,P5	2
P4	9	P3,P8	1	P2,P3,P6	3	P1,P3,P4,P6	3
P5	8	P4,P5	4	P2,P3,P7	1	P1,P3,P5,P6	1
P6	7	P4,P6	6	P2,P3,P8	1	P1,P3,P6,P7	1
P7	1	P5,P6	3	P2,P4,P5	2	P1,P4,P5,P6	2
P8	1	P6,P7	1	P2,P4,P6	3	P2,P3,P4,P5	1
P1,P2	10	P1,P2,P3	6	P2,P5,P6	2	P2,P3,P4,P6	2
P1,P3	10	P1,P2,P4	4	P2,P6,P7	1	P2,P3,P5,P6	1
P1,P4	8	P1,P2,P5	5	P3,P4,P5	2	P2,P3,P6,P7	1
P1,P5	7	P1,P2,P6	4	P3,P4,P6	3	P2,P4,P5,P6	2
P1,P6	6	P1,P2,P7	1	P3,P5,P6	1	P3,P4,P5,P6	1
P1,P7	1	P1,P2,P8	1	P3,P6,P7	1	P1,P2,P3,P4,P5	1
P1,P8	1	P1,P3,P4	5	P4,P5,P6	3	P1,P2,P3,P4,P6	2
P2,P3	6	P1,P3,P5	5	P1,P2,P3,P4	2	P1,P2,P3,P5,P6	1
P2,P4	4	P1,P3,P6	4	P1,P2,P3,P5	3	P1,P2,P3,P6,P7	1
P2,P5	5	P1,P3,P7	1	P1,P2,P3,P6	3	P1,P2,P4,P5,P6	2
P2,P6	4	P1,P3,P8	1	P1,P2,P3,P7	1	P1,P3,P4,P5,P6	1
P2,P7	1	P1,P4,P5	3	P1,P2,P3,P8	1	P2,P3,P4,P5,P6	1
P2,P8	1	P1,P4,P6	5	P1,P2,P4,P5	2	P1,P2,P3,P4,P5,P6	1
P3,P4	5	P1,P5,P6	2	P1,P2,P4,P6	3		

In the second phase, the algorithm computes the support counts (c_k) of all the frequent patterns on the set from the first transaction to the transaction just before the time period T_ξ. Then it computes the milestones of P_k (1≤k≤n) within the range T_ξ. At each valid milestone $\xi^{c_k}(P_k)$ during the scan, it calculates the support of P_k before $\xi^{c_k}(P_k)$ i.e., $sup_{-}^{c_k}(P_k)$ and support of P_k after $\xi^{c_k}(P_k)$ i.e., $sup_{+}^{c_k}(P_k)$. If both of them are greater than t_s, the algorithm then checks the transitional ratio of P_k. If the ratio is greater than t_t then P_k is a positive transitional pattern. Then the algorithm checks whether the transitional ratio of P_k is greater than the current maximal transitional ratio of P_k. If yes, the set of significant frequency ascending milestones of P_k is set to contain $\{\xi^{c_k}(P_k), tran^{c_k}(P_k)\}$ as its single element. If not but it is equal to the current maximal transitional ratio of P_k, $\{\xi^{c_k}(P_k), tran^{c_k}(P_k)\}$ is added into significant frequency ascending milestone of P_k. Similarly it computes negative transitional patterns and their significant milestones if $tran^{c_k}(P_k) \leq -t_t$.

Table 3. Set of Positive transitional patterns and their ascending milestones.

Pattern	Ascending Milestone
P3	aug2006(62.5%), 60.000 %
P4	mar2006(31.25%), 72.500 %
P6	apr2006(37.5%), 72.222 %
P1,P3	aug2006(62.5%), 60.000 %
P1,P4	mar2006(31.25%), 68.571 %
P1,P6	apr2006(37.5%), 66.667 %
P3,P4	may2006(43.75%), 67.857 %
P3,P5	aug2006(62.5%), 60.000 %
P3,P6	may2006(43.75%), 57.143 %
P4,P6	apr2006(37.5%), 66.667 %
P1,P3,P4	may2006(43.75%), 67.857 %
P1,P3,P5	aug2006(62.5%), 60.000 %
P1,P3,P6	may2006(43.75%), 57.143 %
P1,P4,P6	apr2006(37.5%), 58.333 %

Table 4. Set of Negative transitional patterns and their descending milestones.

Pattern	Descending Milestone
P2	may2006(43.75%), -66.667 %
P6	sep2006(68.75%), -63.333 %
P1,P2	may2006(43.75%), -66.667 %
P1,P6	sep2006(68.75%), -56.000 %
P2,P4	may2006(43.75%), -74.074 %
P2,P5	apr2006(37.5%), -60.000 %
P4,P6	aug2006(62.5%), -66.667 %
P1,P2,P4	may2006(43.75%), -74.074 %
P1,P2,P5	apr2006(37.5%), -60.000 %
P1,P4,P6	aug2006(62.5%), -58.333 %
P2,P4,P6	may2006(43.75%), -61.111 %
P1,P2,P4,P	may2006(43.75%), -61.111 %

It was observed that the frequent patterns which contains items P7 or P8 (i.e.,24 patterns) are not used in the second phase of the algorithm, because these items are not contained in the transactions whose position satisfying T_{ξ}. Therefore generating these 24 patterns in the first phase of the algorithm is useless. So, we are proposing a method for pruning these useless frequent patterns in the first phase of the algorithm.

4. ETP-MINE (EFFICIENT TP-MINE) ALGORITHM

In this section, we are presenting an Efficient TP-Mine algorithm which eliminates the useless frequent patterns. The algorithm efficiently generates positive and negative transitional patterns and their significant milestones with respect to a pattern support threshold and transitional pattern threshold.

Algorithm: ETP-Mine

Input:
A transaction database (D), an appropriate milestone range that the user is interested (T_ξ), pattern support threshold (t_s), and transitional pattern threshold (t_t).

Output:
The set of transitional patterns (S_{PTP} and S_{NTP}) with their significant milestones.

Method:

```
Generate frequent 1-itemsets ($L_1$) and their supports from D using the candidate
1-itemsets which appears in the transactions whose position satisfying $T_\xi$
Extract frequent patterns, $P_1, P_{2,}P_3, \dots, P_n$, and their supports using $L_1$ with
min_sup = $t_s$.
Scan the transactions from the first transaction to the last transaction before $T_\xi$ to
compute the support counts, $c_k$ (1 ≥k ≥ n), of all the n frequent patterns on this
part of the database.
$S_{PTP}$ =∅, $S_{NTP}$=∅
for all k = 1 to n do
    MaxTran ($P_k$) = 0, MinTran ($P_k$) = 0
    $S_{FAM}$ ($P_k$) =∅, $S_{FDM}$ ($P_k$) =∅
end for
for all transactions $T_i$ whose position satisfying $T_\xi$ do
    for k = 1 to n do
        if $T_i \supseteq P_k$ then
            $c_k = c_k + 1$
            if $sup_{-}^{c_k}(P_k) \geq t_s$ and $sup_{+}^{c_k}(P_k) \geq t_s$ then
                if $tran^{c_k}(P_k) \geq t_t$ then
                    if $P_k \notin S_{PTP}$ then
                        Add $P_k$ to $S_{PTP}$
                    end if
                    if $tran^{c_k}(P_k)$ > MaxTran ($P_k$) then
                        $S_{FAM}$ ($P_k$) = {$\xi^{c_k}(P_k), tran^{c_k}(P_k)$}
                        MaxTran ($P_k$) = $tran^{c_k}(P_k)$
                    else if $tran^{c_k}(P_k)$ = MaxTran ($P_k$) then
                        Add {$\xi^{c_k}(P_k), tran^{c_k}(P_k)$} to $S_{FAM}$ ($P_k$)
                    end if
                else if $tran^{c_k}(P_k) \leq -t_t$ then
                    if $P_k \notin S_{NTP}$ then
                        Add $P_k$ to $S_{NTP}$
                    end if
```

```
                        if $tran^{c_k}(P_k)$< MinTran $(P_k)$ then
                            $S_{FDM}(P_k) = \{\xi^{c_k}(P_k), tran^{c_k}(P_k)\}$
                            MinTran $(P_k)$ = $tran^{c_k}(P_k)$
                        else if $tran^{c_k}(P_k)$ = MinTran $(P_k)$ then
                            Add $\{\xi^{c_k}(P_k), tran^{c_k}(P_k)\}$ to $S_{FDM}$ $(P_k)$
                        end if
                    end if
                end if
        end if
    end for
end for
return $S_{PTP}$ and $S_{FAM}$ $(P_k)$ for each $P_k \in S_{PTP}$
return $S_{NTP}$ and $S_{FDM}$ $(P_k)$ for each $P_k \in S_{NTP}$
```

By executing this algorithm on the transaction database given in the Table 1, the number of frequent patterns generated is reduced to 63, which are shown in Table 5 and the transitional patterns shown in Table 3 and Table 4 are generated(no change in the transitional patterns).

FP	Sup	FP	Sup	FP	Sup	FP	Sup	FP	Sup
P1	15	P2,P5	5	P1,P3,P5	5	P3,P5,P6	1	P2,P3,P4,P6	2
P2	10	P2,P6	4	P1,P3,P6	4	P4,P5,P6	3	P2,P3,P5,P6	1
P3	10	P3,P4	5	P1,P4,P5	3	P1,P2,P3,P4	2	P2,P4,P5,P6	2
P4	9	P3,P5	5	P1,P4,P6	5	P1,P2,P3,P5	3	P3,P4,P5,P6	1
P5	8	P3,P6	4	P1,P5,P6	2	P1,P2,P3,P6	3	P1,P2,P3,P4,P5	1
P6	7	P4,P5	4	P2,P3,P4	2	P1,P2,P4,P5	2	P1,P2,P3,P4,P6	2
P1,P2	10	P4,P6	6	P2,P3,P5	3	P1,P2,P4,P6	3	P1,P2,P3,P5,P6	1
P1,P3	10	P5,P6	3	P2,P3,P6	3	P1,P2,P5,P6	2	P1,P2,P4,P5,P6	2
P1,P4	8	P1,P2,P3	6	P2,P4,P5	2	P1,P3,P4,P5	2	P1,P3,P4,P5,P6	1
P1,P5	7	P1,P2,P4	4	P2,P4,P6	3	P1,P3,P4,P6	3	P2,P3,P4,P5,P6	1
P1,P6	6	P1,P2,P5	5	P2,P5,P6	2	P1,P3,P5,P6	1	P1,P2,P3,P4,P5,P6	1
P2,P3	6	P1,P2,P6	4	P3,P4,P5	2	P1,P4,P5,P6	2		
P2,P4	4	P1,P3,P4	5	P3,P4,P6	3	P2,P3,P4,P5	1		

Table 5. Frequent Patterns generated using ETP-Mine algorithm

5. EXPERIMENTAL RESULTS

To demonstrate the efficiency of the ETP-Mine algorithm we have done experiments on synthetic data obtained from a Grocery shop with 25 transactions and 16 items. Table 6 shows the comparison of TP-Mine algorithm and our ETP-Mine algorithm against the Grocery data.

Table 6: Comparison of TP-Mine and ETP-Mine algorithms.

	Number of frequent patterns gener in	
Min_su	**TP-Mine**	**ETP-Mine**
1	141	114
2	79	76

6. CONCLUSIONS

The existing TP-Mine algorithm generates more number of frequent patterns which are not useful in generating transitional patterns and their milestones. In this paper we presented a modification to the existing TP-Mine algorithm, which eliminates useless frequent patterns for generating transitional patterns. Our experimental results showed that the proposed algorithm is highly efficient and scalable.

REFERENCES

[1] R.C. Agarwal, C.C. Aggarwal, and V.V.V. Prasad, “Depth First Generation of Long Patterns,” Proc. Sixth ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD ’00), pp. 108-118, 2000.

[2] 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.

[3] R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules,” Proc. 20th Int’l Conf. Very Large Data Bases, pp. 487-499, 1994.

[4] R. Agrawal and R. Srikant, “Mining Sequential Patterns,” Proc.11th Int’l Conf. Data Eng., pp. 3-14, 1995.

[5] S. Brin, R. Motwani, and C. Silverstein, “Beyond Market Baskets: Generalizing Association Rules to Correlations,” Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’97), pp. 265-276, 1997.

[6] D. Burdick, M. Calimlim, J. Flannick, J. Gehrke, and T. Yiu, “Mafia: A Maximal Frequent Itemset Algorithm,” IEEE Trans.Knowledge and Data Eng., vol. 17, no. 11, pp. 1490-1504, Nov. 2005.

[7] J. W. Han, J. Pei, and X. F.Yan, “From Sequential Pattern Mining to Structured Pattern Mining: A Pattern-Growth Approach,” J. Computer Science and Technology, vol. 19, no. 3, pp. 257-279, 2004.

[8] J.W. Han, J. Pei, Y. W. Yin, and R.Y. Mao, “Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach,” Data Mining and Knowledge Discovery, vol. 8, no. 1, pp. 53-87, 2004.

[9] B. Liu, W. Hsu, and Y.-M. Ma, “Integrating Classification and Association Rule Mining,” Proc. Fourth ACM SIGKDD Int’l Conf.Knowledge Discovery and Data Mining (KDD ’98), pp. 80-86, 1998.

[10] N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, “Discovering Frequent Closed Itemsets for Association Rules,” Proc. Seventh Int’l Conf. Database Theory (ICDT ’99), pp. 398-416, 1999.

[11] R.J. Bayardo, Jr., “Efficiently Mining Long Patterns from Data-bases,” Proc. 1998 ACM SIGMOD Int’l Conf. Management of Data (SIGMOD ’98), pp. 85-93, 1998.

[12] R. Srikant and R. Agrawal, “Mining Generalized Association Rules,” Future Generation Computer Systems, vol. 13, nos. 2/3,pp. 161-180, 1997.

[13] Q. Wan and A. An, “Transitional Patterns and Their Significant Milestones,” Proc. Seventh IEEE Int’l Conf. Data Mining, 2007.

[14] Q. Wan and A. An, “Discovering Transitional Patterns and Their Significant Milestones in Transaction Databases,” IEEE Trans. on Knowledge and Data Engg., vol.21,No.12 pp.1692-1707, Dec 2009.

[15] M.J. Zaki and K. Gouda, “Fast Vertical Mining Using Diffsets,” Proc. Ninth ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining (KDD ’03), pp. 326-335, 2003.

B. Kiran Kumar received M.C.A. from Kakatiya University in 1998, and he is pursuing M.Tech in Computer Science and Engg. in JNTU, Hyderabad. He is working as Associate Professor in the department of M.C.A. He delivered guest lectures in the field of Data mining at various engineering colleges. His research interest includes Data mining. He is a member of ISTE.

A.Bhaskar received M.C.A. from Kakatiya University in 1998, and he is pursuing M.Tech in Computer Science and Engg. in JNTU, Hyderabad. He is working as Associate Professor in the department of M.C.A. He delivered guest lectures in the field of Artificial Intelligence at various engineering colleges. His research interest includes Data mining. He is a member of ISTE.