Mining Minimal Infrequent Intervals

D. I. Mazumder
Department of IT
Ibri College of Technology
Ibri, Oman

D. K. Bhattacharyya
Dept. of Computer Science
Tezpur University
Assam, india

M. Dutta
Dept. of CSE
IIIT Guwahati
Assam, india

ABSTRACT

Many real world data are closely associated with the interval of time and distance. Mining infrequent intervals from such data allows users to group transactions with less similarity while mining frequent interval allows user to group the transaction with a similarity above a certain measure. In [1], the notion of mining maximal frequent interval in either a discrete domain or continuous domain is introduced. This paper presents an effective minimal infrequent interval finding algorithm (MII) based on two maximal frequent interval finding techniques represented in [1] and [2] the proposed MII has been established to be effective both theoretically and experimentally.

General Terms

Data mining , I-Tree, Pre-Order Traversal (PT) algorithm.

Keywords

Data mining, maximal frequent interval, minimal infrequent interval, minimum support, discrete and continuous domain.

1. INTRODUCTION

Data mining has received considerable attention during the past few years due to the accumulation of large amounts of data evolving in time, and the need of utilizing these data for analysis. Many techniques and applications have been developed, and among them, mining frequent and infrequent patterns is often used to discover the patterns existing in the data. The most notable applications are association rules, sequential patterns etc. in [4], [5], [6], [7] and [8]. Most of the real world data however, are associated with duration events instead of point events. A record in such data typically consists of the starting time and the ending time. A transaction with starting time‘s’ and ending time‘e’ supports an interval $[a, b]$ only if $s \le a$ and $b \le e$. If the number of transactions supporting the interval $[a, b]$ exceeds a predefined threshold, then $[a, b]$ is called a frequent interval and an interval is said to be infrequent if it is not frequent. Mining infrequent intervals can be applied to many areas. As for an example, a cellular phone company records the time and length of each phone call for billing purposes. Mining infrequent intervals from such data allows the company to discover the intervals during which very few numbers of users are making phone calls and provides information about the degree of accessibility of the network for the customers. As a result the cellular phone company can take a decision about the next status of the channel (ie. running / idle) for those particular periods or even can share the channel with some other networks.

As another example, a web based learning system records the times at which each student logs on and off the system. Mining infrequent intervals enables the system to discover the intervals during which a very few number of students are online. The system may provide this information to the students especially to the research scholars for downloading and uploading their important documents to promote research in respective area of interest. In the examples above, mining infrequent intervals plays an important role on the overall data mining process.

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 defines the problem of MII. Section 4 includes some properties of maximal frequent intervals and minimal infrequent intervals. Section 5 proposes our MII algorithm. Section 6 represents the experimental result and section 7 concludes the paper.

2. RELATED WORKS

In [1] the notion of frequent and maximal frequent intervals for a given database of intervals is defined and an algorithm is presented for the determination of the maximal frequent interval. The algorithm consists of two stages. First the database is scanned once to count the frequency of each interval occurring in the database. These interval frequencies are stored in a so-called I-Tree. Second a Pre-Order Traversal (PT) algorithm is used to discover all maximal frequent intervals. The worst case complexity for constructing the I-Tree is $O(nx(|L| + |R|))$and that for the PT algorithm is $O(max\{nx(|L| + |R|), |L|x|R|\})$, where $|L|$ and $|R|$ are the number of distinct left, right end points of the given intervals respectively. The worst space complexity is $O(|L|x|R|)$. In [2] an improved method of constructing the I-Tree was given which has the worst case time complexity $O(n \log n)$. This was a considerable improvement since in general $\log n$ is far less than $|L| + |R|$. In the tests with experimental data with synthetic dataset as mentioned in [1], the construction of I-Tree has taken the most amount of time in the whole algorithm. Therefore the method in [2] leads to a considerable improvement in the amount of time taken in the execution of the entire algorithm.

In [3] the notion of minimal infrequent interval was defined for multidimensional intervals. It was shown that the problem of generating all minimal infrequent multidimensional intervals can be solved in Quasi-Polynomial time. In this paper as shown in the following sections, we have given a method of obtaining the minimal infrequent intervals for a given database of intervals, after obtaining the maximal frequent intervals. The work in [1] and [2] are for databases of intervals in single dimension only. As mentioned in [1] extension to multi-dimensional intervals will be done in future. Therefore we have also used intervals in single dimension only which have large number of applications as discussed in section [1].

3. PRELIMINARIES

Let DB be a database of ‘n’ transactions t_i for $1 \le i \le n$, where each transaction ‘t’ contains an interval $[l_t, r_t]$ over a discrete domain. Let l_{min} denote the smallest left end point and r_{max} denote the largest right end point among all the intervals occurring in the transactions in DB. For given a transaction *'t'* and an interval $[a,b]$ we introduce the following definitions.

3.1 Definition 1:

A transaction $'t'$ supports $[a,b]$ if $[a,b] \subseteq [l_t, r_t]$ ie if $l_t \le a \le b \le r_t$. For a given interval $[a,b]$, $sup([a,b])$ will denote the number of transactions in DB that supports $[a,b]$.

3.2 Definition 2:

For a given support threshold min_sup with $0 < min\_sup < n$ an interval is called frequent if its support is $\ge \ min\_sup$. Obviously if $[l,r]$ is frequent, then $l_{min} \le l \le r \le r_{max}$. It is also clear that if $[l,r] \subseteq [l',r']$ then $[l,r]$ is frequent if $[l',r']$ is frequent.

3.3 Definition 3:

A maximal frequent interval is a frequent interval which is not properly contained in any frequent interval ie. if an interval $[\boldsymbol{l},\boldsymbol{r}]$ is a maximal frequent interval and $[\boldsymbol{l},\boldsymbol{r}] \subset [\boldsymbol{l}',\boldsymbol{r}']$ then $[\boldsymbol{l}',\boldsymbol{r}']$ is not frequent.

3.4 Definition 4:

An interval $[\boldsymbol{l},\boldsymbol{r}]$ will be called infrequent if $\boldsymbol{l_{min}} \le \boldsymbol{l} \le \boldsymbol{r} \le \boldsymbol{r_{max}}$ and it is not frequent.

3.5 Definition 5:

A minimal infrequent interval is an infrequent interval which does not properly contain any infrequent interval ie. if an interval $[l,r]$ is minimal infrequent and $[l',r'] \subset [l,r]$ then $[l',r']$ is not infrequent.

The problem of mining maximal frequent interval is to discover all the frequent intervals that are maximal and the problem of mining minimal infrequent intervals is to discover all the infrequent intervals that are minimal. Below is an example.

Example1: Consider the database shown in Table 1. Suppose that we only consider the intervals whose end points are in the discrete domain $D = \{v | 1 \le v \le 13$, and v is an integer$\}$. If min_sup is 4, then clearly the maximal frequent intervals are $[2,4], [7,10]$ and $[8,11]$ and the minimal infrequent intervals are $[1,1], [5,5], [6,6], [7,11], [12,12]$ and $[13,13]$.

Table 1. Database for Example 1

Tid	1	2	3	4	5	6	7	8
lti,rti]	[1,5]	[1,4]	[2,6]	[2,5]	6,10]	[6,11]	[7,12]	[8,12]

Based on these definitions we prove some properties using which the proposed minimal infrequent interval finding algorithm has been developed. Next we report those properties

4. SOME PROPERTIES OF THE MAXIMAL FREQUENT AND MINIMAL FREQUENT INTERVALS

In this section we prove certain properties of the minimal infrequent intervals in terms of the maximal frequent intervals. We first note that if $[l,r]$ and $[l',r']$ be two distinct maximal frequent intervals then either (i) $l < l'$ and $r < \ r'$ or (ii) $l' < l$ and $r' < r$. This is so because otherwise one of them will be properly contained in the other contradicting the fact that both are maximal frequent. The same kind of statement will be true for minimal infrequent intervals also. Because of this for any $'a'$ with $l_{min} \le a \le r_{max}$ there can be at most one maximal frequent or minimal infrequent interval with *'a'* as a left end point. Hence the maximal frequent intervals $[l_1,r_1]$, $[l_2,r_2], \dots, [l_k,r_k]$ can be so arranged such that $l_1, < l_2 < \dots < l_k$ and $r_1 < r_2 < \dots < \ r_k$. The following theorem gives an important connection between frequent and maximal frequent intervals. The result is stated and used in [1] without proof.

4.1 Theorem 1:

Every frequent interval is contained in some maximal frequent interval.

Proof: Let $[l,r]$ be a frequent interval. If $[l,r]$ is not maximal frequent then $[l,r] \subset [l',r']$ where $[l',r']$ is frequent. If $[l',r']$ is not maximal frequent, $[l',r'] \subset [l'',r'']$ where $[l'',r'']$ is frequent. Since there are only a finite number of possible intervals with end points between l_{min} and r_{max} , the above process cannot continue indefinitely and we will get some maximal frequent interval $[l_i,r_i]$ such that $[l,r] \subset [l_i,r_i]$. Hence every frequent interval is contained in some maximal frequent interval.

We can now prove the following theorems. □

4.2 Theorem 2:

For every $'\boldsymbol{a}'$ such that $\boldsymbol{l_{min}} \le \boldsymbol{a} \le \boldsymbol{l_1}$, $[\boldsymbol{a},\boldsymbol{a}]$ is a minimal infrequent interval.

Proof: Since $[a,a]$ is not contained in any maximal frequent interval $[l_i,r_i]$ for $1 \le i \le k, [a,a]$ is not frequent. Also it does not properly contain any interval. Hence $[a,a]$ is a minimal infrequent interval □

The proof of the following theorem is similar.

4.3 Theorem 3:

For every $'a'$ such that $r_k < a \le r_{max}$ $[a,a]$ is a minimal infrequent interval.

4.4 Theorem 4:

If $r_i < l_{i+1} - 1$ then for any 'a' with $r_i < a < l_{i+1}$, $[a,a]$ is a minimal infrequent interval.

Proof: Since $a > r_i$, $[a,a]$ is not contained in any $[l_i, r_j]$ for $j \le i$ and since $a < l_i + 1, [a,a]$ is not contained in any $[l_i, r_j]$ for $j \ge i + 1$. Thus $[a,a]$ is not contained in any maximal frequent interval. Hence $[a,a]$ is not frequent and since it does not properly contain any interval, by Definition 5 it is a minimal infrequent interval □

4.5 Theorem 5:

If $r_i \ge l_{i+1} - 1$ then the interval $[l_{i+1} - 1, r_i + 1]$ is a minimal infrequent interval.

Proof: Since $r_i + 1 > \ r_i$, $r_i + 1 > \ r_j$ for all $j \le i$ and hence $[l_{i+1} - 1, r_i + 1]$ cannot be contained in $[l_j, r_j]$ for any $j \le i$. Since $l_{i+1} - 1 < l_i, l_{i+1} - 1 < l_j$ for any $j \ge i$ and hence $[l_{i+1} - 1, r_i + 1]$ cannot be contained in $[l_i, r_j]$ for $j \ge i$. Thus $[l_{i+1} - 1, r_i + 1]$ is not contained in any maximal frequent interval and hence it is not frequent. Suppose $[l_{i+1} - 1, r_i + 1]$ is not a minimal infrequent interval. Then it properly contains an infrequent interval $[a,b]$. If $a > l_{i+1} - 1$, then $[a,b] \subseteq [l_{i+1}, r_i + 1]$. But $r_i < r_{i+1}$ and therefore $r_i + 1 \le r_{i+1}$. Hence $[a,b] \subseteq [l_{i+1}, r_{i+1}]$ and hence is frequent. This implies $a = l_{i+1} - 1$. But in this case we must have $b < r_i + 1$ since $[a,b]$ is properly contained in $[l_{i+1} - 1, r_i + 1]$. Thus $b \le r_i$ and since $l_i < l_i + 1$, we have $l_i \le l_{i+1} - 1 = a$. Therefore $[a,b] \subseteq [l_i, r_i]$ and hence is frequent. This contradicts the fact that $[a, b]$is infrequent. Hence $[l_{i+1} - 1, r_i + 1]$ is a minimal infrequent interval.

In the next section, we propose an algorithm for finding all the minimal infrequent intervals, which simply obtains the intervals as given by Theorem 2,3,4 and 5. For a proof of the completeness of this procedure, we introduce the following lemmas and completeness Theorem 6.

4.6 Lemma1:

If $l_k \le l \le r_k$, there is no minimal infrequent interval with 'l' as the left end-point.

Proof: Suppose there is a minimal infrequent interval $[l, r]$ with $l_k \le l \le r_k$. We cannot have $r \le r_k$ because then $[l, r] \subseteq [l_k, r_k]$ and hence will be frequent. If $r > r_k$ then $[l, r]$ will properly contain $[r, r]$ and by definition of infrequent interval $r \le r_{max}$. Hence $[r, r]$ is infrequent by Theorem2 and thus $[l, r]$ cannot be a minimal infrequent interval which is a contradiction. This proves the lemma,

4.7 Lemma 2:

If $i < k, r_i < l_{i+1} - 1$ and $l_i \le l \le r_i$ then there is no minimal infrequent interval with left end point 'l'.

Proof: Here $i < k, r_i < l_{i+1} - 1$ and $l_i \le l \le r_i$ and let $[l, r]$ be any minimal infrequent interval with left end point l. if $r \le r_i$ then $[l, r] \subseteq [l_i, r_i]$ and hence is frequent. If $r > r_i, [l, r]$ properly contains $[r_i + 1, r_i + 1]$. But $r_i + 1$ satisfies $r_i < r_i + 1 < l_{i+1}$. Hence by Theorem 3 $[r_i + 1, r_i + 1]$ is infrequent and therefore $[l, r]$ is not minimal infrequent. Thus there is no minimal infrequent interval with left end point 'l'.

4.8 Lemma 3:

If $i < k, r_i \ge l_{i+1} - 1$ and $l_i \le l < l_{i+1} - 1$ then there is no minimal infrequent interval with left end point 'l'.

Proof: Let $[l, r]$ be an interval with left end point l. If $r \le r_i$ then $[l, r] \subseteq [l_i, r_i]$ and hence frequent. If $r > r_i$, $[l, r]$ properly contains $[l_{i+1} - 1, r_i + 1]$ which is infrequent according to Theorem 4. Thus $[l, r]$ cannot be minimal infrequent and hence there is no minimal infrequent interval with left end point l.

□

The following theorem now characterizes all the minimal infrequent intervals.

4.9 Theorem 6:

The minimal infrequent intervals given in Theorems 2, 3, 4 and 5 completely determine all the minimal infrequent intervals.

Proof: We have already noted that for any $'a'$ with $l_{min} \le a \le r_{max}$, there is at most one minimal infrequent interval with left end-point $'a'$. If $a < l_1$ then there is one minimal infrequent interval given by Theorem 2. If $a > r_k$ then there is one minimal infrequent interval given by Theorem 3. If $l_k \le a \le r_k$ then there is no minimal infrequent interval with left end point $'a'$ as proved in Lemma 1. That leaves us to examine the case $l_i \le a < l_{i+1}$ for $i = 1,2, \dots k - 1$. For $l_i \le a < l_{i+1}$, if $r_i < l_{i+1} - 1$, then there is no minimal infrequent interval with left end point $'a'$ for $l_i \le a \le r_i$ by Lemma 2 and one minimal infrequent interval with left end point $'a'$ for every $'a'$ satisfying $r_i < a < l_{i+1}$, by Theorem 4. If on the other hand $r_i \ge l_{i+1} - 1$ there is no minimal infrequent interval with left end point $'a'$ for $l_i \le a < l_{i+1} - 1$ by Lemma 3 and one minimal infrequent interval with left end point $l_{i+1} - 1$ by Theorem 5. This covers all the possible values for $'a'$ and hence provides the proof of completeness

5. THE PRPOSED MINIMAL INFREQUENT INTERVAL (MII) ALGORITHM

At first in the first part of the algorithm the maximal frequent intervals are determined using algorithms given in [1] and [2]. After that in the second part, the following algorithm determines the minimal infrequent intervals by scanning all the possible left end points of such intervals which range from l_{min} to r_{max} and identifying those that are given by Theorems 2,3,4 and 5. During the execution of the algorithm, a set of intervals M is maintained which is initially empty and the minimal infrequent intervals are inserted one by one into M as those are determined. In the first phase of the second part, for every value of $'a'$ such that $l_{min} \le a < l_1$, we insert an interval $[a, a]$ (by Theorem 2).

In the second phase of the second part, for $i = 1,2, \dots, k - 1$ we insert intervals into M as follows

if $r_i < l_{i+1} - 1$ then for every $'a'$ such that $r_i < a < l_{i+1}$, the interval $[a, a]$ is inserted into M (by Theorem 4).

else we insert $[l_{i+1} - 1 , r_i + 1]$ into M (by Theorem 5).

In the last phase of the second part for every $'a'$ such that $r_k < a \le r_{max}$, we insert $[a, a]$ into M (by Theorem 3).

The soundness of MII follows from Theorems 2, 3, 4 and 5 and its completeness is established in Theorem 6.

5.1 MII algorithm

MII is executed in the second part, which accepts the maximal frequent intervals $[l_1, r_1]$, $[l_2, r_2], \dots, [l_k, r_k]$ with $l_1, < l_2 < \dots < l_k$ determined in first part by the algorithms given in [1] and [2] based on the data set DB. Next we report the pseudo code of MII.

```
 MII(DB)
 {
M=empty.
For $a = l_{min}$ to $l_1 - 1$
      M ← M∪ {$[a, a]$}   //by Theorem 2
For $i = 1$, to $k - 1$
  if ($r_i < l_{i+1} - 1$)
   {
     for $a = r_i + 1$ to $l_{i+1} - 1$
     M=M∪ {$[a, a]$}        //by Theorem 4
   }
  else
     M=M∪ {$[l_{i+1} - 1, r_i + 1]$} //by Theorem 5
 For $a = r_k + 1$ to $r_{max}$
    M=M∪ {$[a, a]$}           //by Theorem 3
return M
}
```

Fig: 1 Pseudo code of MII

5.2 Complexity analysis of the algorithm of MII

After the maximal frequent intervals are determined, the minimal infrequent intervals are determined one by one, each in $O(1)$ time. Since each minimal infrequent interval must have a distinct left end point, there cannot be more than $r_{max} - l_{min} + 1$ of them. Without loss of generality using a translation of the intervals we can take $l_{min} = 1$. Hence the complexity for the determination of the minimal infrequent intervals is $O(r_{max})$ which is $O(n)$ when r_{max} is $O(n)$. This is in addition to the time required to determine the maximal frequent intervals. Once the maximal frequent intervals are determined, the rest of the work can be completed in the memory since r_{max} is usually quite small and no further database scan is required.

The space complexity for this additional work is $O(r_{max})$ since we simply need the space for the arrays containing the maximal frequent and minimal infrequent intervals which are at most r_{max} in number.

6. EXPERIMENTAL RESULTS

Following environment and datasets were used to test the effectiveness of MII experimentally.

6.1 Environment Used

MII was implemented in a workstation having Intel(R) Xeon(R) CPU X5470 of speed 3.33GHz, with a 4 GB RAM in a Linux environment using C++.

6.2 Dataset Used

To test the correctness of MII we use synthetic data generator. We developed a data generator based on [1] and generated several data sets (as reported in the first column of Table 2) by varying the number of records. For each data set, the number of transactions is n, the left end points of the intervals are distributed uniformly between 1 and l_{max}, and the length of these intervals is in Poisson distribution with mean m. Minimum support is measured in percentage of n.

6.3 Results

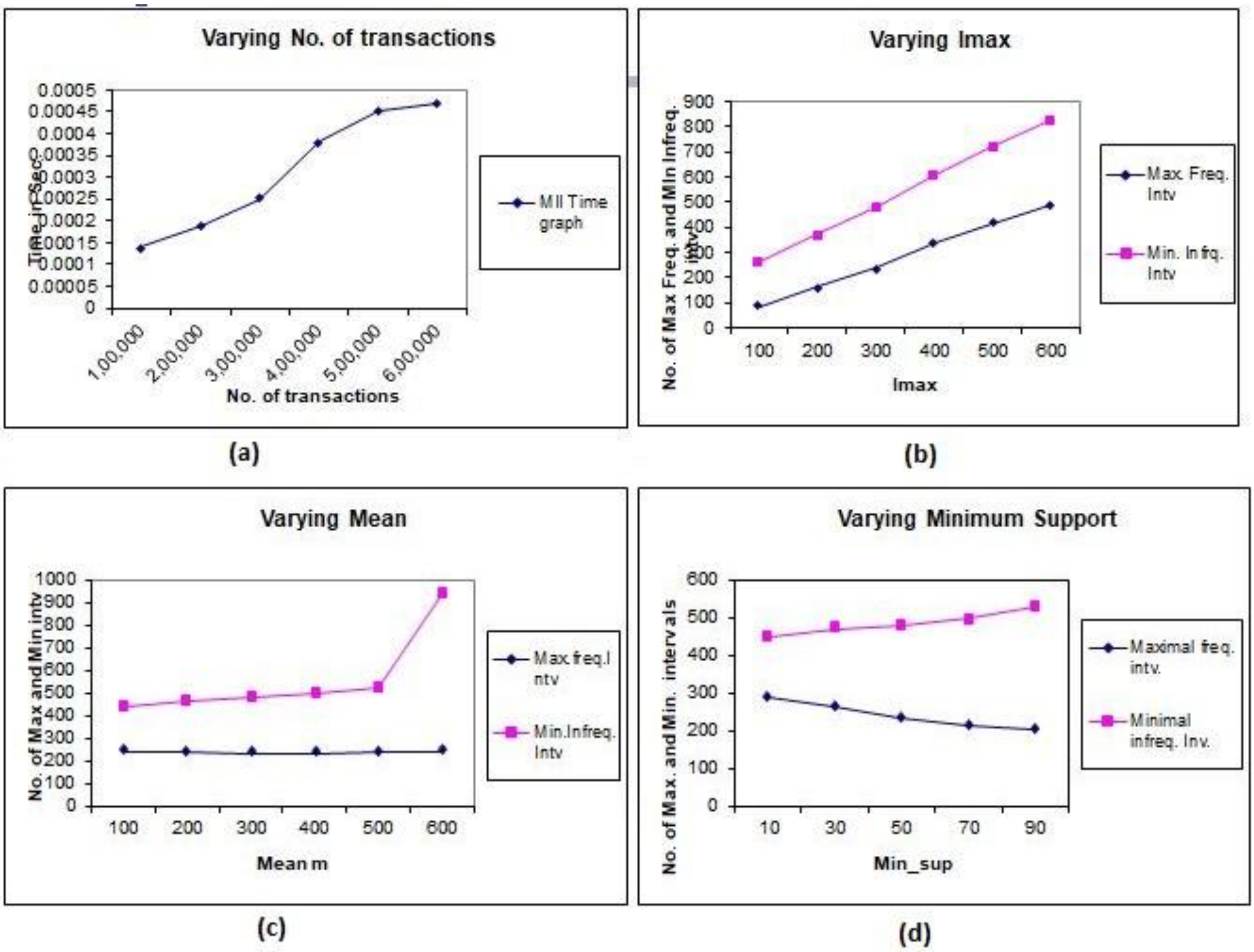
The algorithm is experimentally tested with fixed values of l_{max} and m. Size of the data set is varied and the identification of number of frequent and infrequent intervals are recorded. The running time of the algorithm also noted with the variation of the size of the dataset. It is observed that the running time of the algorithm is directly proportional to the size of the dataset irrespective of the presence of number of frequent and infrequent intervals. The observations are listed in the Table 1.

The graphical representation of the results are shown in fig. 2 in which four different variations are considered as explained below:

In fig. 2(a), the graph is drawn between number of transactions and time taken in seconds. It is observed from the graph that with increase of number of transactions, time consumption also increased. Similarly, in fig. 2(b), the graph is drawn by varying Imax with number of maximal frequent and minimal infrequent intervals. The graph in fig. 2(c) represents the effect of the algorithm by varying the values of mean and in fig. 2(d), the graph signifying the behavior of the algorithm by varying minimum support.

Table 1. Experimental results over synthetic dataset

Dataset Used	Description	No. of records	No. of maximal frequent intervals	No. of minimal frequent intervals	Time in seconds
100klmax_200m_100	lmax=200, Mean m=100	1,00,000	169	296	0.000139s
200klmax_200m_100	lmax=200,Mean m=100	2,00,000	175	302	0.000188s
300klmax_200m_100	lmax=200, Mean m=100	3,00,000	173	309	0.000225s
400klmax_200m_100	lmax=200, Mean m=100	4,00,000	172	307	0.000381s
500klmax_200m_100	lmax=200, Mean m=100	5,00,000	173	309	0.000454s



(a) (b) (c) (d)

Fig: 2 Graphical representations of the results of MII

7. CONCLUSIONS AND FUTURE WORK

An effective method of mining infrequent interval (MMI) is reported in this paper. The effectiveness of MII is established theoretically as well as experimentally. $O(n)$ time complexity and $O(r_{max})$ space complexity makes the algorithm more effective. There is a possibility that the mining infrequent interval can be extended for a continuous domain also. Work is going on towards the extension of MII for handling continuous valued data for finding minimal infrequent intervals.

8. REFERENCES

[1] Lin, J. (2003). Mining Maximal Frequent Intervals. In the Proceedings of ACM symposium on Applied Computing, pp. 426-431. ACM, New York.

[2] Dutta, M. and Mahanta, A.K. 2010. An efficient method for construction of I-tree. In: Proceedings of National workshop on Design and Analysis of Algorithm(NWDA).

[3] Elbassioni, Khaled M. Finding All Minimal Infrequent

Multi-dimensionalIntervals:Max-Planck-Institutf¨ur Informatik, Saarbr¨ucken, Germany.

[4] Agrawal, Rakesh and Srikant, Ramakrishnan 1994. Fast Algorithms for Mining Association Rules in Large Databases, Proceedings of the 20th International Conference on Very Large Data Bases, p.487-499, September 12-15.

[5] Agrawal, Rakesh and Srikant, Ramakrishnan 1995. Mining Sequential Patterns, Proceedings of the Eleventh International Conference on Data Engineering, p.3-14, March 06-10.

[6] Lu, H., Han, J. and Feng, L 1998. Stock movement and n-dimensional inter-transaction association rules. In Proc. 1998 SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery (DMKD'98).

[7] Liu, B., Hsu, W., and Ma Y. 1998. Integrating classification and association rule mining. In KDD p. 80-86.

[8] Mannila, Heikki and Toivonen, Hannu 1997. Levelwise Search and Borders of Theories in KnowledgeDiscovery, Data Mining and Knowledge Discovery, v.1 n.3, p.241-258, [doi>10.1023/A:1009796218281]

Authors Biographies

D. I. Mazumder, received his Master Degree in Mathematics from Gauhati University, India in 2003. He received his M.Tech in Information Technology, from Tezpur University, India in 2010. Currently He is working as a Lecturer of Mathematics in the Dept. of Information Technology in IBRI College of Technology, OMAN. His major Research interests include Data mining and Cryptography.

D. k. Bhattacharyya, received his MCA from Dibrugarh University, India in 1991. He received PhD Degree in Computer Science from Tezpur University, India in 1999. Presently he is working as a Professor in the Department of Computer Science and Engg, Tezpur University, India. His Major research interests include Cryptography, Error correction/ detection, Content based image retrieval, Data mining and Network Security.

M. Dutta, received his M.Sc. Degree in Physics from Delhi University in 1972. He received PhD Degree in Mathematics from IIT Kanpur in 1979. He also received M. S. in Computer Science from University of Houston in1982. Presently he is working as a professor in the Department of Computer Science and Engineering, Tezpur university, India. His major research area of interest are P=NP problem, Optimization and Data mining.