Mining Rare Periodic-Frequent Patterns Using Multiple Minimum Supports

by

R. Uday kiran, P Krishna Reddy

in

15th International Conference on Management of Data
(COMAD2009)

Report No: IIIT/TR/2009/261

Centre for Data Engineering
International Institute of Information Technology
Hyderabad - 500 032, INDIA
December 2009
Mining Rare Periodic-Frequent Patterns Using Multiple Minimum Supports

R. Uday Kiran P. Krishna Reddy
Center for Data Engineering
International Institute of Information Technology-Hyderabad
Hyderabad, India - 500032.
uday_rage@research.iiit.ac.in and pkreddy@iiit.ac.in

Abstract

Recently, an approach has been proposed in the literature to extract frequent patterns which occur periodically. In this paper, we have proposed an approach to extract rare periodic-frequent patterns. Normally, the single minsup based frequent pattern mining approaches like Apriori and FP-growth suffer from “rare item problem”. That is, at high minsup, frequent patterns consisting of rare items will be missed, and at low minsup, number of frequent patterns explode. In the literature, efforts have been made to extract rare frequent patterns under “multiple minimum support framework” [5, 6, 7, 8].

Several extensions to frequent pattern mining have been investigated in the literature. Recently, an approach has been proposed to extract new kind of frequent patterns, called periodic-frequent patterns [4]. It considers both the items of a transaction along with the timestamp (occurrence time), and extracts knowledge pertaining to periodic occurrence of patterns.

In this paper, we have made an effort to extract rare periodic-frequent patterns. The existing periodic-frequent pattern mining approach also suffers from the “rare item problem” as it is also based on single minsup constraint [4]. In this paper, we have developed a new algorithm to extract rare periodic-frequent patterns. Experiment results show that the proposed approach is efficient.

1 INTRODUCTION

It can be observed that most of the data mining approaches discover the knowledge pertaining to frequently occurring entities. However, real-world datasets are mostly non-uniform in nature containing both frequent and relatively infrequent or rarely occurring entities. In literature, it has been reported that rare knowledge patterns i.e., knowledge pertaining to rare entities may contain interesting knowledge useful in decision making process [1]. The rare knowledge patterns are more difficult to detect because they present in fewer data cases. In the literature, research efforts are being made to investigate efficient approaches to extract rare knowledge patterns like rare association rules and rare class identification [1].

Frequent pattern mining is an important model in data mining. Normally, the single minsup based frequent pattern mining approaches like Apriori [2] and FP-growth [3] suffer from “rare item problem”. That is, at high minsup, frequent patterns consisting of rare items will be missed, and at low minsup, number of frequent patterns explode.

In the literature, efforts have been made to extract rare frequent patterns under “multiple minimum support framework” [5, 6, 7, 8].

Several extensions to frequent pattern mining have been investigated in the literature. Recently, an approach has been proposed to extract new kind of frequent patterns, called periodic-frequent patterns [4]. It considers both the items of a transaction along with the timestamp (occurrence time), and extracts knowledge pertaining to periodic occurrence of patterns.

In this paper, we have made an effort to extract rare periodic-frequent patterns. The existing periodic-frequent pattern mining approach also suffers from the “rare item problem” as it is also based on single minsup constraint [4]. In this paper, we have developed a new algorithm to extract rare periodic-frequent patterns by extending “multiple minimum support framework”. Experimental results on synthetic and real-world datasets show that the proposed approach is efficient.

The paper is organized as follows. In Section 2, we discuss the background information about the “rare item problem” in mining frequent patterns and “multiple minimum support framework”. We also discussed the “rare item problem” in periodic-frequent pattern mining approach. In Section 3, we discuss the proposed approach. Experimental results conducted on synthetic and real world datasets are presented in Section 4. Section 5 concludes the paper.

2 Background

2.1 Overview of Frequent Pattern Mining

Frequent patterns are important class of regularities which exist in a database. The basic model of frequent patterns is as follows [2]:

Let \( I = \{i_1, i_2, \ldots, i_n\} \) be a set of items. Let \( T \) be a set of transactions (dataset), where each transaction \( t \) is a set of items such that \( t \subseteq I \). A pattern (or an itemset) \( X \) is a set of items \( \{i_1, i_2, \ldots, i_k\} \), where \( 1 \leq k \leq n \) such that \( X \subseteq I \).
Support of $X$, denoted as $S(X)$, refers to the proportion of transactions that contain $X$ in the transaction dataset. The pattern $X$ is a frequent pattern, if $S(X) \geq \text{minsup}$, where $\text{minsup}$ refers to user-specified minimum support.

### Example 1 (Running Example)
Consider the set of items $I = \{\text{bread, ball, pen, jam, bat, bed, pillow, book}\}$ shown in Table 1. The dataset contains 10 transactions. Let $\text{minsup} = 0.4$. The pattern $\{\text{bread, jam}\}$ is one of the frequent pattern because its support is 0.4, which is equal to $\text{minsup}$.

### 2.2 Rare Item Problem in Frequent Pattern Mining
Rare items are the items having low support values. Rare frequent patterns are the frequent patterns consisting of only rare items, or both frequent and rare items. With a single $\text{minsup}$ constraint, mining of rare frequent patterns causes the following problem. At high $\text{minsup}$, frequent patterns involving rare items will be missed, and at low $\text{minsup}$ number of frequent patterns will be exploded. This dilemma is called “rare item problem” [5].

### Example 2: We now explain the “rare item problem” by considering two $\text{minsup}$ values for the dataset shown in Table 1. If $\text{minsup} = 0.4$, the set of frequent patterns generated is $\{\{\text{bread}, \{\text{ball}\}, \{\text{jam}\}, \{\text{bat}\}, \{\text{bread, jam}\}\}\}$. It can be observed that the rare pattern $\{\text{bed, pillow}\}$ is missed. If $\text{minsup} = 2$, then the set of frequent patterns generated is $\{\{\text{bread}, \{\text{ball\}, \{\text{jam\}, \{\text{bat\}, \{\text{bread, jam\}, \{\text{bat, ball\}, \{\text{bed, pillow\}}\}}\}\}\}$. It can be observed that even though the missed rare pattern $\{\text{bed, pillow}\}$ is extracted at low $\text{minsup}$ value, however the number of frequent patterns is increased.

### 2.3 Multiple Minimum Support Framework
In the literature, efforts are being made to extract rare frequent patterns under “multiple minimum support framework” [5, 6, 7, 8]. The basic idea is as follows. To extract frequent patterns only a single $\text{minsup}$ is used for entire dataset, whereas under “multiple minimum support framework,” each item is specified with a different minimum support value, called minimum item support ($\text{MIS}$). Rare frequent patterns are extracted by considering $\text{MIS}$ values for all items. Under multiple minimum support framework, $\text{minsup}$ for a pattern is calculated using Equation 1.

$$\\text{minsup}(i_1, i_2, \ldots, i_k) \geq \min \left( \frac{\text{MIS}(i_1), \text{MIS}(i_2), \ldots, \text{MIS}(i_k)}{\text{MIS}(i_k)} \right)$$

where, $\text{minsup}(i_1, i_2, \ldots, i_k)$ represents the support of a pattern $\{i_1, i_2, \ldots, i_k\}$ and $\text{MIS}(i_j)$ represents the minimum item support for the item $i_j \in I$.

### Example 3: For the dataset of Table 1, let the user-specified $\text{MIS}$ values for the items bread, ball, pen, jam, bat, bed and pillow be 4, 4, 3, 3, 2 and 2 (in counts) respectively. The frequent patterns generated are $\{\{\text{bread}\}, \{\text{ball}\}, \{\text{jam}\}, \{\text{bat}\}, \{\text{bed}\}, \{\text{pillow}\}, \{\text{bread, jam}\}, \{\text{bat, ball\}, \{\text{bed, pillow\}}\}$.

### 2.4 Overview of Periodic-Frequent Pattern Mining
Periodic-Frequent patterns are special kind of frequent patterns which occur periodically (or regularly) within a dataset [4]. In this approach, the time of occurrence of each transaction is taken into account for periodic-frequent pattern mining.

So, in the periodic-frequent pattern mining, the $\text{tid}$ represents timestamp of a particular transaction $t$. Let $\text{tid}_d$ be the $\text{tid}$ in which the pattern $X$ has occurred. Suppose, a pattern $X$ appears in two transactions, say $t_d^1$ and $t_d^2$, where $i$ and $j$ are $\text{tids}$ and $i < j$ (i occurred before j). A period of pattern $X$, say $p^X$, is the difference between $t_d^j - t_d^i$. Let, $p^X$ denote the set of all periods of $X$ i.e., $p^X = \{p^X_1, \ldots, p^X_o\}$. Then, $\text{Per}(X) = \max(p^X_1, \ldots, p^X_o)$ is called the periodicity of pattern $X$. A pattern is said to be periodic-frequent, if its periodicity is no greater than the user-specified maximum periodicity ($\text{maxper}$) constraint and its support is no less than the user-given minimum support ($\text{minsup}$) constraint.

### Example 4: For the dataset of Table 1, we assume that the first column represents the timestamps of the transactions. The pattern $\{\text{bread, jam}\}$ has occurred in $t_1, t_4, t_5$ and $t_9$. A period for this pattern, $p^{\text{bread, jam}}$ by considering the $t_1$ and $t_4$ is 3 ($4 - 1$). Similarly, the other periodicities are 1 (5-4) and 4 (9-5). The periodicity of the pattern $\{\text{bread, jam}\}$, $\text{Per}(\{\text{bread, jam}\}) = \max(3,1,4) = 4$. Let the user-specified $\text{minsup}$ and $\text{maxper}$ values are 4 and 4 respectively. Then, this pattern is a periodic-frequent pattern because $S(\text{bread, jam}) \geq \text{minsup}$ and $\text{Per}(\text{bread, jam}) \leq \text{maxper}$.

Conventional frequent pattern mining approaches like Apriori or FP-growth cannot be used for mining periodic-frequent patterns because they do not capture the periodicity of the patterns. Therefore, another pattern-growth approach has been discussed in [4]. In this approach, a tree, called Periodic-Frequent tree (PF-tree) is constructed and mined using conditional patterns (base) to discover complete set of periodic-frequent patterns.

### 2.5 Rare Item Problem in Periodic-Frequent Pattern Mining
As the basic model of periodic-frequent patterns uses only single $\text{minsup}$ as one of its constraint, this model also suffers from the dilemma called “rare item problem” similar to the “rare item problem” in frequent pattern mining approaches such Apriori and FP-growth.
3 Mining Rare Periodic-Frequent Pattern

3.1 Basic Idea

To extract rare periodic-frequent patterns, the “multiple minimum support framework” has to be extended to the basic model of periodic-frequent patterns. Under “multiple minimum support framework”, the MIS values for the items are taken into account to extract rare frequent patterns. Similarly, if we extended “multiple minimum support framework” to the existing approach of periodic-frequent patterns for extracting rare periodic-frequent patterns, the existing approach has to be modified by taking into account the MIS values of the items.

For extending multiple minimum support framework to the existing periodic-frequent pattern approach, the following issues are to be addressed.

- Criteria for specifying minsup for a pattern.
- Extending the existing Periodic Frequent-tree (PF-tree) approach discussed in [4] to mine rare periodic-frequent patterns.

3.1.1 Criteria for specifying minsup for a pattern

The input to the proposed approach is set of transactions containing items, MIS values for the items and maximum periodicity (maxper).

Under “multiple minimum support framework” for frequent patterns, the criterion for specifying minsup for a pattern is shown in Equation 1. Similarly, for “multiple minimum support framework” based periodic-frequent pattern mining, the minsup for a pattern is shown in Equation 2.

\[ S(i_1, i_2, ..., i_k) \geq \min \left( MIS(i_1), MIS(i_2), ..., MIS(i_k) \right) \]  

\[ Per(i_1, i_2, ..., i_k) \leq \text{maxper} \]

where \( S(i_1, i_2, ..., i_k) \) represents the support of a pattern \( \{i_1, i_2, ..., i_k\} \), \( Per(i_1, i_2, ..., i_k) \) represents the periodicity of a pattern \( \{i_1, i_2, ..., i_k\} \) and MIS(\( i \)) represents the minimum item support for the item \( i \in I \).

3.1.2 Extending “multiple minimum support framework” to PF-tree

The existing PF-tree approach is summarized as follows. The PF-tree contains two components (i) PF-list and (ii) prefix-tree. PF-list is a list which is used to maintain each item’s frequency and periodicity values. Prefix-tree is a tree similar to the FP-tree [3]; however, instead of containing support of an item at each node, the tid of the transaction is maintained at the leaf node of the corresponding item. The corresponding tid list of the item’s nodes is called tid-list of that item.

The construction of PF-tree is as follows. The PF-list is constructed with the initial scan on the dataset. From the PF-list, items which have support less than minsup and periodicity greater than maxper are pruned. Next, items in the PF-list are sorted in descending order of their support values. Using, the sorted list of items, prefix-tree is constructed by performing another scan on the dataset. Using conditional pattern bases, the constructed PF-tree is mined to discover complete set of periodic-frequent patterns.

We now address the issues regarding the extensions to the PF-tree approach to mine rare periodic-frequent patterns. First, we discuss about downward closure property and sorted closure property of the patterns.

According to downward closure property, all non-empty subsets of frequent patterns are frequent.

According to sorted closure property, if a sorted pattern \( \langle i_1, i_2, ..., i_k \rangle \), for \( k \geq 2 \) and MIS(\( i_1 \)) \leq MIS(\( i_2 \)) \leq \ldots \leq MIS(\( i_k \)), is frequent then all its subsets consisting of the item having lowest MIS value i.e., \( i_1 \) must be frequent. However, the other subsets need not necessarily be frequent. (The sorted closure property has been elaborately discussed in [5] [8]).

It was observed that the patterns mined using maxper constraint follow downward closure property. Whereas, the patterns mined using “multiple minimum support framework” follow sorted closure property.

The issue here is which property do the rare periodic-frequent patterns follow. It can be noted that the rare periodic-frequent patterns follow sorted closure property. As a result, to extract rare periodic-frequent patterns we have to consider all subsets of a pattern for generating rare periodic-frequent patterns.

The other issue is construction of PF-tree (PF-list and prefix-tree) by considering MIS values of the items. To extract rare periodic-frequent patterns, we store the item’s MIS values by creating additional column in the PF-list. And, the prefix-tree has to be constructed with MIS descending order of items.

3.2 Description of the Proposed Approach

The input to the proposed approach is the transactional dataset (along with the timestamp of each transaction), item’s MIS values and maxper constraint. The output is the complete set of periodic-frequent patterns. The approach involves three steps: Construction of MIS-PF-tree, Construction of compact MIS-PF-tree and Mining Patterns from the compact MIS-PF-tree. We briefly discuss these steps.

3.2.1 Construction of MIS-PF-tree

The MIS-PF-tree is a tree consisting of two components: MIS-PF-list and prefix-tree. The MIS-PF-list is a list which contains four fields - item name (i), minimum item support (MIS), total support (f) and periodicity of item i (p). The structure of the prefix-tree used in MIS-PF-tree is same as the prefix-tree used in PF-tree [4]. However, items are arranged in support descending order in the prefix-tree of the PF-tree, whereas the items are arranged in MIS descending order in the prefix-tree of MIS-PF-tree.

We define the following variables. Let \( id_t \) be the temporary array which records the tids of last occurring trans-
actions of the items in the MIS-PF-list. Let $t_{cur}$ and $p_{cur}$ denote the tid of current transaction and the most recent period for an item. The construction of MIS-PF-tree is as follows. Before scanning the transaction dataset, the MIS-PF-list is constructed by adding the items in descending order of their MIS values and by setting $f = 0$ and $p = 0$. In the prefix-tree, a root node is created and labeled as “null”. The $id_l$ value of every item are set to 0. Let the ordered list of items in the MIS-PF-list be $L$.

Next, in each transaction $t_{cur}$, identify the items in it. For each item in the respective transaction, perform the following steps in the MIS-PF-list: (i) increment $f$ by 1 (ii) calculate $p_{cur} = t_{cur} - id_l$ and update $p = p_{cur}$, if $p_{cur} > p$ (iii) set $id_l = t_{cur}$. For the same transaction a branch is created in the prefix-tree and the tid of the respective transaction is added to tid-list of the tail node represented by this transaction. (The creation procedure of a branch in the prefix-tree is same as that in PF-tree; however, we do not maintain the frequency value at each node.) After scanning all the transactions, update the $p$ value of every item in the MIS-PF-list by calculating the $p_{cur}$ value with $t_{cur}$ equivalent to the tid of the last transaction in the transaction dataset. To facilitate tree traversal, an item header table is built so that each item points to its occurrences in the tree via a chain of node-links.

We explain the construction of MIS-PF-tree by considering the dataset shown in Table 1. The initial MIS-PF-tree is shown in Figure 1 (a). The MIS-PF-tree generated after scanning first, second and the last transaction are shown in Figure 1(b), Figure 1(c) and Figure 1(d). Figure 1(e) shows the updated MIS-PF-list after scanning the complete transaction dataset.

### 3.2.2 Constructing the compact MIS-PF-tree

The items in the MIS-PF-tree can be pruned using the following observations.

- If the support of an item is less than the lowest MIS value among all periodic-frequent items, such item will not generate any periodic-frequent pattern.
- If the periodicity of an item is greater than $\maxper$, such item will not generate any periodic-frequent pattern.

In the MIS-PF-list, we collect the set of all periodic-frequent items and identify the lowest MIS value among them. Let us call this value as periodic lowest minimum item support (PLMIS). In the MIS-PF-list, items which have $\text{support} < \text{PLMIS}$ or periodicity $> \maxper$ are identified and pruned from the MIS-PF-tree one after the other because they do not generate any periodic-frequent pattern. The procedure followed for pruning each item from the MIS-PF-tree is as follows. In the MIS-PF-list, completely remove the respective item from the list. In the prefix-tree, perform tree-pruning operation to remove all nodes of the respective item. If a pruned node has a $\text{tid-list}$ then $\text{tid-list}$ is transferred to its respective parent node.

After tree-pruning operation, the resultant prefix-tree may have parent nodes, where each parent node may have multiple child nodes of a same item. Therefore, tree-merging operation is performed on the prefix-tree to merge such child nodes. The tree-merging operation involves generating a common branch by merging the $\text{tid-lists}$ of different child nodes having same item. The resultant MIS-PF-tree derived after tree-pruning and tree-merging operations is called compact MIS-PF-tree. The MIS-PF-list in the compact MIS-PF-tree is called compact MIS-PF-list.

Continuing with the example, the periodic-frequent items in the MIS-PF-list are bread, ball, jam, bat, bed and pillow. The lowest MIS value among all these periodic-frequent items is 2. Therefore, using $\text{PLMIS} = 2$ and $\maxper = 4$, we prune the items book and pen from the MIS-PF-list and prefix-tree of MIS-PF-tree. Figure 2(a) and Figure 2(b) show the MIS-PF-tree generated after
pruning item ‘book’ and item ‘pen’. It can be observed in Figure 2(b) that there exists two different branches, one with $\text{tid-list} = 2$ and another with $\text{tid-list} = 6$, sharing same set of items. So we perform tree-merging operation to merge such branches. The compact MIS-PF-tree derived after tree-merging operation is shown in Figure 3(a).

![Figure 2: MIS-PF-tree. (a) After pruning item book and (b) After pruning item pen.](image)

### 3.2.3 Mining the compact MIS-PF-tree

Mining the compact MIS-PF-tree is similar to the mining of PF-tree [4]. However, the difference is we use the MIS value of the suffix item as the $\text{minimum support}$ for all of its conditional pattern bases.

Continuing with the example, mining the periodic-frequent patterns from the compact MIS-PF-tree shown in Figure 3(a) is as follows. The item ‘pillow’ which is at the bottom-most of the MIS-PF-list is initially chosen for mining periodic-frequent patterns. Its prefix-tree, MIS-PF$_{pillow}$, shown in Figure 3(b) is constructed with the prefix sub-paths of nodes labeled with item ‘pillow’. From MIS-PF$_{pillow}$, the conditional-tree, MIS-CT$_{pillow}$ (see Figure 3(c)) is constructed by removing all non-periodic frequent nodes. From MIS-CT$_{pillow}$, the pattern {bed, pillow} is generated as periodic-frequent pattern because $S(\text{bread, jam}) \geq \text{MIS}(\text{pillow})$ and $P(\text{bread, jam}) \leq \text{maxper}$.

To enable construction of prefix-tree for the remaining items in the compact MIS-PF-list, the tid-lists of the item ‘pillow’ are pushed-up to respective parents nodes in the original compact MIS-PF-tree and all nodes of item ‘pillow’ in compact MIS-PF-tree are deleted thereafter. Similarly process is carried out for remaining items in the compact MIS-PF-tree. Finally, the periodic-frequent patterns generated are $\{\{\text{bread}\}, \{\text{ball}\}, \{\text{jam}\}, \{\text{bat}\}, \{\text{bed}\}, \{\text{pillow}\}, \{\text{bread, jam}\}, \{\text{bat, ball}\}, \{\text{bed, pillow}\}\}$.

### 4 Experimental Results

In this section, we present the performance comparison of the proposed approach with the approach discussed in [4]. We have evaluated the performance results pertaining to the generation of periodic-frequent patterns by considering two kinds of datasets: synthetic and real world datasets. The synthetic dataset T10.I4.D100K, is generated with the data generator [2], which is widely used for evaluating association rule mining algorithms. It contains 1.00,000 number of transactions and 886 items. Another dataset is a real world dataset referred as retail dataset [9]. It contains 88,162 number of transactions and 16,470 items.

For our experiments, we used the method discussed in [7] for specifying the MIS values for the items (see Equation 3).

$$\text{MIS}(i_j) = S(i_j) - SD \text{ if } (S(i_j) - SD) > \text{LS}$$

$$= \text{LS} \text{ otherwise}$$

where, $S(i_j)$ refers to support of an item $i_j \in I$, $\text{LS}$ refers to user-specified “least support” value, MIS($i_j$) refers to minimum item support for an item $i_j \in I$, SD refers to user-specified “support difference” value, $\lambda$ represents the parameter like mean, median and mode of the item supports and $\beta \in [0, 1]$.

For both T10.I4.D100k and Retail datasets, the MIS values are calculated by fixing $\text{LS}$ value at 0.1% and varying the $\text{SD}$ values. The $\text{SD}$ values are calculated with $\lambda$ representing the mean of the items having support values no less than the $\text{LS}$ (0.1%) value and $\beta$ varying at 0.2, 0.4, 0.6 and 0.8. The derived $\text{SD}$ values for synthetic and retail datasets are shown in Table 2. For both T10.I4.D100k and Retail datasets, the maximum periodicity (maxper) has been set to 2.5% and 5% respectively.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Synthetic dataset</th>
<th>Retail dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\text{SD} = \lambda(1-\beta)$</td>
<td>$\text{SD} = \lambda(1-\beta)$</td>
</tr>
<tr>
<td>$\beta = 0.8$</td>
<td>0.25%</td>
<td>0.07%</td>
</tr>
<tr>
<td>$\beta = 0.6$</td>
<td>0.50%</td>
<td>0.14%</td>
</tr>
<tr>
<td>$\beta = 0.4$</td>
<td>0.75%</td>
<td>0.21%</td>
</tr>
<tr>
<td>$\beta = 0.2$</td>
<td>1.00%</td>
<td>0.28%</td>
</tr>
</tbody>
</table>

Both Figure 4(a) and Figure 4(b) show how the number of periodic-frequent patterns varied at different $\text{SD}$ values and $\text{maxper} = 0.05\%$. In these figures, the X-axis represents the $\text{SD}$ ($\beta$) value and the Y-axis represents the number of periodic-frequent patterns generated. The think line in this graph represents the number of periodic-frequent patterns generated using the approach discussed in [4], with $\text{minsup} = 0.1\%$ and $\text{maxper} = 0.05\%$. It can be observed that the number of periodic-frequent patterns is significantly reduced by our method with low $\text{SD}$ value. When $\text{SD}$ value becomes larger, the number of periodic-frequent patterns found by our method gets closer to that found by the approach discussed in [4]. The reason is, when $\text{SD}$ becomes larger more and more items’ MIS values reach $\text{LS}$.

Both Figure 4(c) and Figure 4(d) show the runtime taken for generating periodic-frequent patterns at different $\text{SD}$
values and maxper = 0.05%. The think line in this graph represents the runtime of the approach discussed in [4]. It can be observed that as the SD value increases, runtime of the proposed approach also increases and gets closer to that of [4]. The reason is, when SD increases, more number of periodic-frequent patterns are generated.

5 Conclusions and Future Work

In this paper, we have proposed an approach to extract rare periodic-frequent patterns. The existing periodic-frequent pattern mining approach suffer from the “rare item problem” while extracting rare periodic-frequent patterns. We extended multiple minimum support framework which was proposed to extract rare frequent patterns to extract rare periodic-frequent patterns by incorporating the notion of periodicity and proposed a new pattern-growth algorithm to extract rare periodic-frequent patterns. Experiment results on both synthetic and real-world datasets show that the proposed approach is efficient.

In this approach, we have extended “multiple minimum support framework” to only support variable of the periodic-frequent pattern approach. It can be noted that there is a scope to improve the performance by extending “multiple minimum support framework” to the periodicity variable of the periodic-frequent pattern approach. We are planning to investigate it as a part of future work.

References


[3] Jiawei, H., Jian, P., Yiwen, Y., and Runying, M. “Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach*.”, ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, 2004, pp. 53-87.


