MaxRI: A method for discovering maximal rare itemsets

Sadeq Darrab
David Broneske
Gunter Saake
University of Magdeburg
Magdeburg, Germany
firstname.lastname@ovgu.de

ABSTRACT

Rare itemset mining got extensive attention due to its high importance in real-life applications. Rare itemset mining methods aim at discovering the whole set of rare itemsets in a dataset. Although current algorithms perform reasonably well in finding interesting rare itemsets, they also reveal a large number of rare itemsets, including redundant ones. As a result, skimming through these massive amounts of (partly redundant) itemsets is a big overhead in many applications. On the other hand, generating a massive number of rare itemsets also compromises the performance of algorithms in terms of time and memory. To address these limitations, we propose an efficient algorithm called maximal rare itemset (MaxRI) to discover maximal rare patterns (long rare itemset). Then, we propose another method RRI (Recover Rare Itemsets from maximal rare itemsets) to retrieve the interesting subset of rare itemsets of a user-given length, k, from the set of maximal rare itemsets. To the best of our knowledge, this is the first paper proposed for rare itemset mining by considering the representative rare patterns without redundant ones. Our experimental results indicate that our proposed methods' performance is better than the up-to-date algorithms in terms of time and memory consumption.

CCS CONCEPTS

• **Information systems → Association rules**.

KEYWORDS

rare itemsets, representative rare itemsets, maximal rare itemsets

ACM Reference Format:
Sadeq Darrab, David Broneske, and Gunter Saake. 2021. MaxRI: A method for discovering maximal rare itemsets. In *2021 4th International Conference on Data Science and Information Technology (DSIT 2021), July 23–25, 2021, Shanghai, China.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3478905.3478972

1 INTRODUCTION

Rare itemset mining (RIM) [22, 23] provides meaningful information since it discovers uncommon phenomena from a dataset. It is widely used in many real-life applications, such as identifying fraudulent credit card transactions[2], medical diagnosis [3], and adverse drug reactions[4]. For example, in the health care domain, frequently occurred complications may be less interesting than infrequently (unexpected) ones whose early discovery would avoid adverse consequences. In traffic accidents analysis, discovering irregular behaviors leads to knowing the real cause of accidents. Therefore, discovering unusual events (rare itemsets) is more interesting than frequently occurred patterns in many real-life domains.

There is a plethora of methods presented to address the rare itemset problem. These methods can be grouped based on the original method to apriori-based[5], FP-based [9], and N-list-based [10] methods. Apriori-based methods [5, 6] use the most well-known algorithm called an apriori algorithm [1] to mine useful rare itemsets. The apriori-based methods employ a candidate-test fashion to generate the whole set of rare itemsets. They first scan the dataset to retrieve all interesting single items, then proceed to find all 2-itemsets, then 3-itemsets, etc. These methods discover candidate rare itemsets of length k via joining rare itemsets of length k−1. Apriori-based methods inherit the apriori algorithm's shortcomings such as overly massive candidate sets, redundant scans over the dataset, and zero-support candidate itemsets [9]. To overcome these limitations, the FP-tree-based algorithm, RP-growth [9], proposed to mine rare itemsets. The RP-growth algorithm scans datasets at most twice. In the first scan, the frequency of 1-items are calculated. Then, it constructs an RP-tree by adding transactions that have at least one rare item. Although the RP-growth algorithm outperforms the shortcomings of apriori-based algorithms, it generates a vast amount of unnecessary conditional trees that compromises search time and storage consumption.

To avoid these drawbacks, Rare Pre Post (RPP) algorithm was proposed to recover desired rare itemsets [10]. The RPP algorithm generates an RN-list of all interesting rare items. Then, in the mining process, the RN-lists of rare items are utilized to create the whole set of rare itemsets by intersecting operations. Although the RPP algorithm works well for sparse datasets, its performance degrades when mining rare itemsets from dense datasets.

Although current algorithms work well to address the rare item problem, they produce a large number of rare itemsets, including redundant ones. Rare itemset mining is very sensitive to the user-defined support threshold since a very low minimum support threshold yields an intractable number of rare itemsets. Generating a large number of rare itemsets degrades the performance of an algorithm in terms of execution time and memory cost. The performance of an algorithm becomes worse when dealing with dense datasets. Besides, the resulting itemsets have to be further aggregated in an extensive downstream analysis. The situation is

DSIT 2021, July 23–25, 2021, Shanghai, China

ACM ISBN 978-1-4503-9024-8/21/07…$15.00
https://doi.org/10.1145/3478905.3478972

critical while dealing with abnormal events (rare itemsets) since discovering them as early as possible would avoid unfavorable consequences. Therefore, how to design a method for identifying condensed representation of rare itemsets without redundant ones is a critical research topic.

A solution to the mentioned shortcomings is to reveal compact representations of rare itemsets instead of generating the whole set of rare itemsets. This paper proposes a maximal rare itemset algorithm called, MaxRI to discover a compact representation of rare itemsets without redundant ones. Mining representative rare itemsets can reduce computational time and memory consumption and make them easier to be analyzed by an expert. The MaxRI algorithm requires three fundamental phases: 1) in the first phase, unpromising items (i.e., items that do not satisfy a given minimum rare support threshold, minSup) are removed from transactions since they play no role to generate rare itemsets, 2) in the second phase, the MRP-tree (maximal rare itemset tree) is constructed, and 3) in the third phase, the compact representations of rare itemsets is retrieved. This is the first method to discover representative rare itemsets without a candidate-test framework to the best of our knowledge.

Since an expert may be interested in the information of the subset of the resulting maximal rare itemsets, we propose a Recovering Rare Itemsets (RRI) procedure to recover interesting rare itemsets from the set of maximal rare itemsets that satisfy a user-given length, k.

The main contributions of this paper can be summarized as follows.

- An MaxRI algorithm is proposed to discover representative rare itemsets without producing candidate ones as Apriori-based methods do.
- The MaxRI algorithm finds the complete set of representative (maximal) rare itemsets without generating conditional trees as FP-based algorithms do.
- The MaxRI algorithm efficiently recovers the whole set of maximal rare itemsets directly from the tee.
- Several experiments are conducted on real dense datasets to evaluate the performance of the proposed algorithm. Experimental results show that the MaxRI algorithm is orders of magnitude faster than the state of art algorithms.
- The Recovery Rare Itemset algorithm, RRI, is presented to generate the set of interesting rare itemsets (of a fixed length k set by the user) from the collection of maximal rare itemsets.
- From the resulting maximal rare itemses, we can find the information of the complete set of rare itemses without redundant ones.

The remainder of this paper is structured as follows. In Section 2, the background and preliminaries of rare itemset mining are presented. In Section 3, we introduce the literature review. The proposed method with a motivating example are presented in section 4. Section 5 shows the experimental results. Finally, we conclude the paper and highlight the future work in section 6.

Table 1: Original dataset

TID	Items	Sorted items
1	1, 3, 4	3, 1, 4
2	2, 3, 5	2, 3, 5
3	1, 2, 3, 5	2, 3, 5,1
4	2, 5	2, 5
5	1, 2, 3, 5	2, 3, 5, 1

2 BACKGROUND AND PRELIMINARIES.

To understand the basic concepts of rare itemset mining, let us consider the following motivating example.

Motivating Example: given a transaction dataset DB in Table 1, let maximum support threshold (maxSup) and rare support threshold (minSup) be 0.80 and 0.20, respectively. The task of rare itemset mining is to extract the set of rare itemsets with support greater than or equal to minSup and less than maxSup.

Table 2: Support of 1-items.

Item	Support
5	4
3	4
2	4
1	3
4	1

Table 3: Tidset of items

Item	Tidset
1	T1, T3, T5
2	T2, T3, T4, T5
3	T1, T2, T3, T5
4	T1
5	T2, T3, T4, T5

Definition 2.1 Relative support of an itemset X: Given an itemset $X = \{x_1, x_2, ..., x_n\}$ and the dataset DB in Table 1, the support of the itemset X is the number of transactions that contain X such that

$$Sup(X) = \frac{\mid \{T \in DB \mid x_1 \in T \wedge x_2 \in T \wedge ... \wedge x_n \in T\} \mid}{\mid DB \mid}$$

For example the support of the itemset = {2, 5} is 3/5 since it occurs in transactions 2, 3, and 5.

Definition 2.2 Rare itemsets [22]: An itemset X is called rare itemset if its relative support, denoted as Sup (X), in a given dataset is less than the frequent support threshold, freqSup, such that Sup(X) < freqSup. For example, let $freqSup = 0.5$, the itemset $X = \{4, 1\}$ is a rare itemset since $Sup(X) = 0.20$ which is less than freqSup.

Definition 2.3 Interesting rare itemsets [5]: A rare itemset X is called interesting if its relative support satisfies the following condition:

$$Sup(X) < maxSup \wedge Sup(X) \geq minSup$$

.

Definition 2.4 Maximal rare itemset: An itemset X is called maximal rare itemset (MRI) iff Definition 2.3 holds and there is no interesting rare itemset in which X is a subset. The complete set of MRIs in the motivating example are $\{(4, 1, 3) : 1, (1, 2, 3, 5) : 2\}$.

In this paper, we focus on mining maximal rare itemsets that satisfy Definitions 2.3, and 2.4. From now on, we will use rare itemsets as a shorthand for interesting rare itemsets.

3 LITERATURE REVIEW

Rare itemset mining gained considerable attention in the last decades because of its importance in real-life domains. These methods can be grouped into two categories based on the exploration traversal of the search space. In the first subsection, the first category, called breadth-first search methods, is presented [5, 6]. In the second subsection, the depth-first search methods are introduced [9, 11].

3.1 Breadth-First Search Methods.

Breadth-first search methods utilize an apriori algorithm [1] to recover the whole set of rare itemsets. These methods discover rare itemsets by joining the candidate of length $k-1$ to generate candidate itemsets of length k. First, they scan the dataset to retrieve all interesting rare 1-itemsets. Then, the candidate itemsets of length two are generated by joining all interesting rare itemsets of length 1. The process is terminated when no further candidate itemsets can be generated.

In [5], an apriori-inverse algorithm is presented to extract rare itemsets with a support value less than a maximum support threshold (maxSup). In the first scan, it finds 1-itemsets whose support value does not satisfy the maxSup. Then, the remaining process is similar to the formal apriori algorithm [1]. In [6], two algorithms are presented to mine interesting rare itemsets. In the first phase, the minimal rare itemsets (itemsets with no rare subset) are generated by an apriori-rare algorithm. An MRG-Exp algorithm then uses the minimal rare itemsets as a seed to generate the whole set of rare itemsets. In [5, 6], the search space is explored in a bottom-up fashion. However, exploring the search space in a bottom-up technique is costly in terms of runtime and memory consumption [12]. Unlike [5, 6], the AfRIM algorithm [12] explores the search space in a top-down fashion. The AfRIM algorithm forms the longest n-itemset that comprises all single items in a dataset. Then, it discovers the candidate n-1-itemset subsets from rare n-itemsets and collects the interesting rare itemsets that satisfy user-defined thresholds. The AfRIM algorithm is expensive in terms of time and memory since it starts with the longest n-itemset that contains zero-itemsets in the mining process. In [13], the rarity algorithm presented to avoid zero-itemset generation since it starts from items in the longest transaction in the dataset. The remaining process is similar to the AfRIM algorithm.

The above-mentioned methods use at most two thresholds to recover the rare itemsets. In [14, 15], the MSapriori algorithm and its optimizations are proposed to discover both frequent and rare itemsets by utilizing a minimum item support (MIS) framework. These methods extract frequent itemsets, including rare ones, by assigning lower support thresholds for the rare (infrequent) itemsets than those for most common (frequent) itemsets. They work similar to the apriori [1] with the following significant difference. The itemset is considered an interesting (frequent and rare) itemset if its support satisfies the lowest MIS of items within it.

Breadth-first search methods inherit apriori algorithm's shortcomings, such as overly massive candidate sets, redundant scans over the data, and zero-support candidate itemsets.

3.2 Level-Depth Exploration Methods.

To address the shortcomings of the breadth-first search methods, level-depth methods such as RP-growth [9], and RPP [10], have been proposed to extract rare itemsets without the expensive itemset generation and pruning steps. In [9], the RP-growth algorithm utilizes a tree data structure, the FP-Tree [16]. This algorithm scans the dataset at most twice. In the first scan, the occurrence count of the 1-itemsets is computed. Then, in the second scan, the RP-Tree is created by transactions with rare items. For the mining process, RP-growth applies a divide-and-conquer strategy to create rare itemsets. Nevertheless, for each rare item, RP-growth generates a conditional tree in the mining process. The RP-growth method becomes costly when datasets are sparse datasets since building conditional trees makes RP-growth inefficient. In [10], the RPP algorithm is proposed to mine rare itemsets from sparse datasets efficiently. It first constructs a tree called an RPPC-tree, to keep all information needed to mine rare itemsets. Then, RN-lists (i.e., maintain information such as count, pre-order, and post-order of each node in the RPPC-tree) of all interesting rare items are created. The RN-lists of rare items are used in the mining process to obtain the whole set of rare itemsets by intersecting these lists.

Apart from single support threshold methods, several methods are proposed to retrieve frequent itemsets, including rare ones [11, 17–19]. In [18, 19], based on the FP-growth algorithm [16], CFP-growth [18], and CFP-growth++ [19] algorithms utilize MIS to discover frequent itemsets including rare itemsets. In these methods, the dataset is scanned once to build a CFP-tree. Then, the tree is reconstructed by employing pruning and merging techniques. However, reconstruction of the tree is costly in terms of time and memory consumption. To fill this gap, the MISFP-growth [11] algorithm has been proposed to extract both frequent and rare itemsets. It efficiently builds the tree without the need for a reconstruction phase. Unlike the FP-growth based methods, mis-eclat[17] utilizes a vertical representation of data to extract the whole set of frequent itemsets, including rare ones.

Although these methods address the rare itemset problem, they suffer from overly huge generated itemsets, including redundant ones. Rare itemset mining is very sensitive to the user-defined support threshold since a very low minimum support threshold yields an intractable number of rare itemsets. The performance of an algorithm becomes worse when dealing with dense datasets. Furthermore, the resulting itemsets have to be further aggregated in an extensive downstream analysis. The situation is critical while dealing with abnormal events (rare itemsets) since discovering them as early as possible would avoid unfavorable consequences. Therefore, designing a method for identifying condensed representation of rare itemsets without redundant ones is a critical research topic. Although there are several methods [20, 21] proposed to extract

maximal and closed frequent itemsets, there is no method proposed for condensed rare itemsets, to the best of our knowledge.

In this paper, we propose an algorithm, MaxRI, which recovers a compressed representation of rare itemsets without redundant ones. Extracting condensed representation(maximal rare) itemsets can reduce computational time and memory consumption. In addition, maximal rare itemsets are easier to analyze by an expert. For further analysis, the RRI algorithm is proposed to mine interesting rare itemsets from the set of maximal rare itemsets that satisfy a user-given length, k.

4 MAXIMAL RARE ITEMSET (MAXRI) ALGORITHM

In this section, we describe the proposed algorithm, MaxRI, which mines a concise representation of rare itemsets (maximal rare itemsets) without unnecessary ones. The MaxRI algorithm aims at extracting representative rare itemsets that can reduce the runtime, memory cost and make them easier to be analyzed by experts. Additionally, we propose the RRI algorithm to find out the set of interesting rare itemsets (of a fixed length k set by the user) from the collection of concise representation of rare itemsets that are generated by the MaxRI algorithm.

The MaxRI algorithm utilizes an FP-tree structure [16] to discover the whole set of long rare itemsets. First, the MaxRI algorithm performs a preprocessing phase to remove unpromising items (i.e., items with support less than minSup threshold). Then, this method compresses the dataset into a compact rare tree structure, MRI-tree (maximum rare itemset tree). Finally, the mining process is run to retrieve representative rare itemsets without the candidate-test fashion and conditional trees. To illustrate how the MaxRI algorithm works, let us show the three phases with the motivating example.

4.1 Preprocessing Phase.

In this phase, the dataset is scanned once to find the support count of 1-items. Then, unpromising items are removed from the dataset. Transactions are sorted in descending order according to their support. The resulting dataset is shown in the right column in Table 1, and the support of the 1-itemset is shown in Table 2. In this phase, we generate Tidset (a set of transaction-ids) to calculate the support of the subset rare itemsets that can be derived from the maximal rare itemsets as shown in Table 3.

4.2 Construction of the MRI-tree.

The MRI-tree is constructed by transactions in the right column in the preprocessed dataset in Table 2. Similar to the FP-tree, the dataset is scanned once again to build the tree. Initially, the tree contains a root labeled as `null`. Next, the transactions in the right column in Table 2 are inserted into the MRI-tree in decreasing order of their support. If the inserted transactions share a prefix with some previously entered ones, then the count of all nodes along the path is increased by 1. Otherwise, new nodes are created and initialized to 1. In the tree, each node consists of a name, count, children, a parent, and a link to similar nodes with the same name. For example, given the node (2:4) in the tree, numbers 2 and 4 represent the node's name and occurrence in a path {1532}, respectively. The node (2:4)

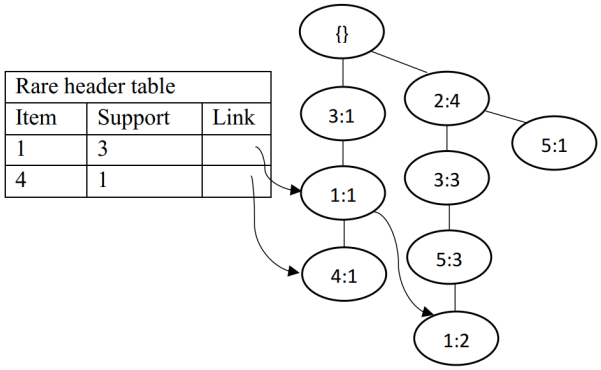

Figure 1: The compact MRI-tree after adding all transactions.

has two children 3, and 5, and its parent is the root of the tree. The resulting compact tree of our motivating example is given in Fig. 1. To facilitate the tree traversal, the rare header table is created to hold only rare items that can be used to generate maximal rare itemsets. Reordering items in decreasing order of support items resulted in a compact tree where the most interesting rare items are placed at the bottom of tree.

4.3 Mining Process.

The mining process follows a bottom-up fashion. It starts from the lowest item in the rare header table. For each rare item, the MaxRI algorithm retrieves corresponding paths for an item i in the compact MRI-tree. The count of a retrieved path is set to the occurrence of item i in this path. The count of prefix items cannot exceed the suffix item's count since items are inserted in descending order of their support.

Assume that SetMRI is the set of all retrieved maximal rare itemsets. For each item i in the rare header table, the mining process can be summarized as follows.

- Case 1: For an item i, there is only one single path, P, generated. First, we check if the occurrence of item i does not satisfy minSup and maxSup thresholds, we overstep mining with item i and continue with the remaining items in the rare header table. Otherwise, we append the item i to the path P and set the support of this path as equals the occurrence of item i. Then, we check if this path is not a subset of any maximal rare itemset in SetMRI, we add the resulting path to SetMRI. In contrast, we discard it.
- Case 2 :For an item i, there are two or more paths generated. Let X be the longest path that is retrieved for item i. In such a case, we have different situations that should be considered as follows.
  (1) For the longest path X, we add it to SetMRI if the occurrence of X satisfies both maxSup and minSup constraints, and there is no maximal rare itemset in SetMRI that contains X. Thus, we add the longest maximal rare itemset as early as possible.
  (2) For each two paths Y and Z, where $Y \subset Z$, the occurrence of Y becomes the sum of Y.count + Z.count. We add Y

to SetMRI if the existence of Y meets both maxSup and minSup thresholds, and there is no maximal rare itemset in SetMRI that contains Y.

(3) For each path K that is not a subset of any other paths, we add it to SetMRI if its occurrence in the tree satisfies both maxSup and minSup constraints, and there is no maximal rare itemset in SetMRI contains K.

(4) The same process is repeated for remaining items in the rare header table.

To illustrate how the mining process works, let us follow our motivating example. Given the compact MRI-tree and rare header table in Fig. 1, the task is to find the whole set of MRIs. In this regard, the MaxRI algorithm starts with the lowest rare item, 4, in the rare header table. For item 4, there is one prefix path from the root to the suffix item 4, which is $\{1:1, 3:1\}$. The number after : represents the count of that item in the path. The only representative itemset that can be generated from item 4 is $\{413:1\}$ since $Sup(413) = 0.20$, which is less than $maxSup = 0.80$ and it is also greater than or equals to $minSup = 0.20$. The maximal rare itemset $\{413:1\}$ is added to the set of representative rare itemsets, SetMRI.

For rare item 1, there are two prefix paths from the root to suffix item 1, $\{5:3, 3:3, 2:4\}$ and $\{3:1\}$. The longest maximal rare itemset is $\{1532\}$ and its support count equals the relative support count of item 1 in these paths, which is 0.40. Thus, $\{1532\}$ with support 0.40 will be added to SetMRI since $\{1532\}$ is not a subset of any maximal rare itemset in SetMRI. For the path $\{31\}$, we discard it since there is a maximal rare itemset ($\{1532\}$) in SetMRI that contain $\{31\}$.

Thus, the concise representations of rare itemsets are $\{413 : 0.20, 1532 : 0.40\}$. These are the representative rare itemset for the whole set of rare itemsets. The resulting rare itemsets are discovered efficiently and they can be easily interpreted by an expert.

For the same dataset in our motivating example with the same maxSup and minSup thresholds, the whole set of rare itemsets that would be generated by traditional methods such as RP-Tree and RPP algorithms are the following: $\{(4:0.20), (41:0.20), (43:0.20), (413:0.20), (1:0.60), (15:0.40), (13:0.40), (12:0.40), (153:0.40), (152:0.40), (132:0.40), (1532:0.40), (53:0.60), (32:0.60), (532:0.60)\}$.

Thus, generating the complete set of rare itemsets by traditional methods not only degrades the performance but also analyzing them is computationally expensive.

4.4 Recovering Rare Itemsets Procedure.

The generated maximal rare itemsets are easily interpreted. However, once they are generated, the support of their subset itemsets is missed. An expert may be interested in some subset of maximal rare itemset for further analysis. We fill this gap by proposing an algorithm called RRI (Recover Rare Itemsets from maximal rare itemsets) to retrieve the interesting subset rare itemset of length k from the set of maximal rare itemsets. The RRI algorithm works as follows. It takes as input the Tidset of items shown in Table 3, which have been generated in the preprocessing phase, the length of desired rare itemsets, k, and the set of maximal rare itemsets with length no less than k. The results of the RRI algorithm are the complete set of rare k-itemsets. For each maximal rare itemset, X, the RRI algorithm intersects Tidset of the rare itemset with the Tidsets of length k-1 of the other items in X.

Following our motivating example, let us show how the RRI algorithm proceeds. The Tidset is given in Table 3, and the desired length of subset rare itemsets is set to 3. There is one maximal rare itemset $\{1532\}$ with a length greater than 3. The Tidset (1) is intersected with each 2-subset of the remaining items (53, 52, 32). The resulting Tidset of the intersection process of 1 and (5, 3) is $T3, T5$. Thus, the relative support of the $\{153\}$ itemset is $|Tidset(153)| = 2/5 = 0.40$. The same process is repeated for the rare item 1 with the remaining subsets. The resulting subset rare itemsets for the rare item 1 are $\{153 : 0.40, 152 : 0.40, 132 : 0.40\}$. The process is terminated since no more rare itemsets can be produced. The results are $\{153 : 0.40, 152 : 0.40, 132 : 0.40\}$.

5 EXPERIMENTAL RESULTS

To evaluate the performance of the proposed methods, MaxRI and RRI, for mining representative rare itemsets, we conducted several experiments on two dense real-world datasets (Mushroom, Accidents) [24]. Table 4 shows the characteristics of these datasets where #Trans and Avg represent the number and average of transactions, respectively. The experiments are conducted on windows 10, 64-bit operating system, Intel Core i7- 7700HQ CPU 2.80 GHz with 16 GB main memory, and 1 TB hard disk. The performance of the proposed algorithms is compared against the state-of-art algorithms [9, 10], RP-growth, and RPP, in terms of execution time and memory cost. To have a common implementation environment, the algorithms are implemented in Java.

Table 4: Dataset characteristics

Name	#Trans	#Item	Avg
Mushroom	8416	119	23
Accidents	340,183	468	33.8

5.1 Execution Time.

To measure the execution time of the proposed methods, MaxRI and RRI, the experimental results are compared with the state-of-art algorithms, RPP, and RP-growth, on the datasets shown in Table 4. To limit the number of generated rare itemsets, the maximum constraint (maxSup) is fixed at 10%, and the minSup value is varied from 0.1% to 1% for all experiments. Thus, the interesting rare itemsets should be less than maxSup and greater or equal to minSup. In each graph, the X-axis represents different values of minSup, whereas the Y-axis stands for the execution time.

For the Mushroom dataset, Fig. 2 shows the runtime of the proposed algorithms MaxRI and RRI, which is compared with RPP and RP-growth algorithms. As it can be seen from Fig. 2, the proposed methods, MaxRI and RRI, outperform the state-of-art methods, RP-growth and RPP for all minSup values. The performance of the proposed methods is a factor of 1000 better than the traditional methods for rare itemsets mining. The advantage of the proposed methods is that MaxRI shrinks the search space efficiently since it does not need to generate any candidate itemsets or projected conditional trees. The MaxRI algorithm recovers representative

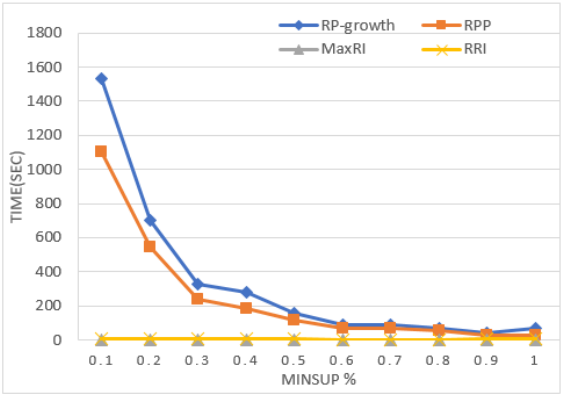

Figure 2: Execution time for Mushroom dataset.

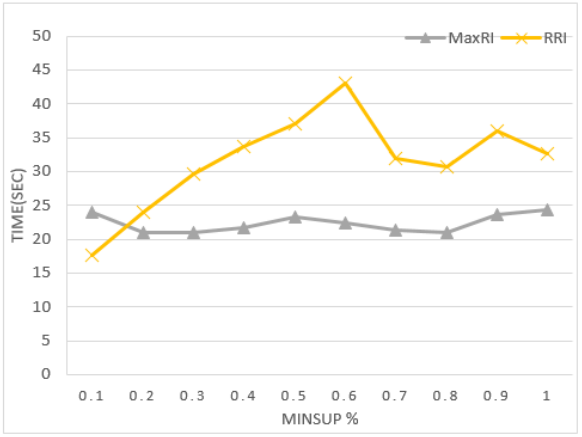

Figure 3: Execution time for Accidents datasets.

rare itemsets (long rare itemsets) without projected conditional trees. The RRI algorithm performs only the necessary interesting operations to discover desired subset itemsets with a specific length. This is because it receives the Tidset of items and the desired maximal rare itemsets. Then, it performs the intersection of a rare item with the remaining items to discover the subset of interesting rare itemsets.

For the Accidents dataset, Fig. 3 shows the runtime of the proposed algorithms MaxRI and RRI. For the competitors' algorithms, RP-growth and RPP, the runtime does not appear in the graph since they could not complete their processing after two hours. Although the maximal constraint is set at 0.1, the execution time for RP-growth and RPP algorithms is not completed. Then, the highest minSup value, 0.01, is chosen to see the runtime for RP-growth and RPP algorithms. The execution time for RP-growth and RPP algorithms was about 1000 sec. Furthermore, they generate 339814644 rare itemsets while mining with minSup = 1%. For the same minSup =1%, the MaxRI generates only 91 representative rare itemsets. This supports our claim that generating representative rare itemsets is more interesting since we should take a rapid response to abnormal events, and they can be interpreted easily. From the graph, we can see that the runtime for MaxRI and RRI algorithms are at most 25 and 43, respectively, which is a huge improvement compared to the state-of-the-art methods.

5.2 Memory Consumption.

To measure the memory consumption of the proposed methods, MaxRI and RRI, the experimental results are compared with the state-of-art algorithms, RPP and RP-growth, on the datasets shown in Table 4. We use the same setup as for the experiment before and show the different values of minSup on the X-axis, while the Y-axis represents the memory cost.

For the Mushroom dataset, Fig. 4 shows the memory consumption of the proposed algorithms MaxRI and RRI compared with RPP and RP-growth algorithms. The proposed methods, MaxRI and RRI, consume less memory than RPP and RP-growth algorithms. We can see that the proposed RRI algorithm consumes slightly more memory than the MaxRI algorithm. This is because the RRI algorithm needs to hold the Tidset of items in memory for the intersection processes. Both RP-growth and RPP algorithms are memory costly since they generate a massive amount of rare itemsets. Besides, RP-growth generates conditional trees that involve more memory. The RPP algorithm consumes more memory than others because it needs to keep the pre-order and post-order nodes in the tree. Also, it keeps the RN-lists of items in memory for intersection operations.

For the Accidents dataset, the memory costs of the proposed methods are illustrated in Fig. 5. As we mentioned in the execution part, the memory cost of the competitors' algorithms, RP-growth and RPP, does not appear in the graph since they could not complete their runtime for two hours. Fig. 5 shows that the proposed RRI algorithm consumes more memory than the proposed method, MaxRI. This is because the RRI algorithm maintains the Tidset of items in memory for the intersection operations.

5.3 Discussion.

Our empirical results indicate that rare itemset mining from dense datasets produces an unmanageable number of rare itemsets. The high number of resulting itemsets leads to an extensive downstream analysis, and the classical methods' performance (i.e., RP-growth and RPP algorithms) is costly in terms of runtime and memory decay. Our proposed methods address these limitations by generating representative rare itemsets faster than compared methods and consuming less memory. As observed from the graphs, the runtime and memory gain of our proposed methods are significantly better for very dense datasets such as Mushroom and Accidents. Fig. 2-5 indicate that the performance of the proposed algorithm is better than the state-of-art algorithms in terms of time and memory costs. The MaxRI algorithm recovers the concise representation of rare itemsets known as maximal rare itemsets. In terms of rare itemsets, the maximal rare itemsets are information lossless. This is due to the fact that the rare item (i.e., in each maximal rare itemsets) has the lowest support count in the retrieving paths. Thus, any subset rare itemset that contains this rare item has support equals to the occurrence of the rare item. For example, in our motivating example, the rare item 1 leads to generate the maximal rare itemsets,{1532 : 0.40}. For any subset of this maximal rare itemset that contains

the rare item 1, its occurrence is 0.40. Therefore, from the resulting representative rare itemsets, we are able to discover meaningful rare itemsets without producing a massive number of meaningless and redundant rare itemsets. We can see that the proposed methods are about a factor of 1000 faster than the compared algorithms.

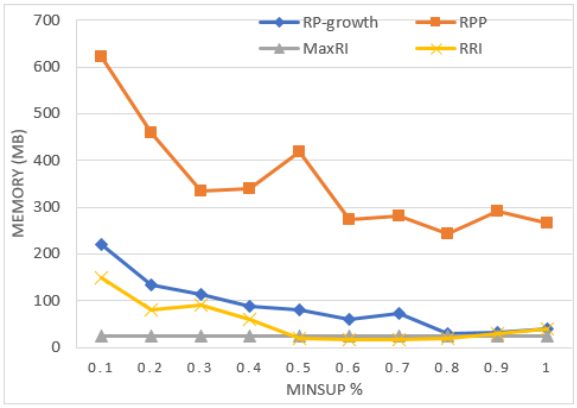

Figure 4: Memory cost for Mushroom datasets.

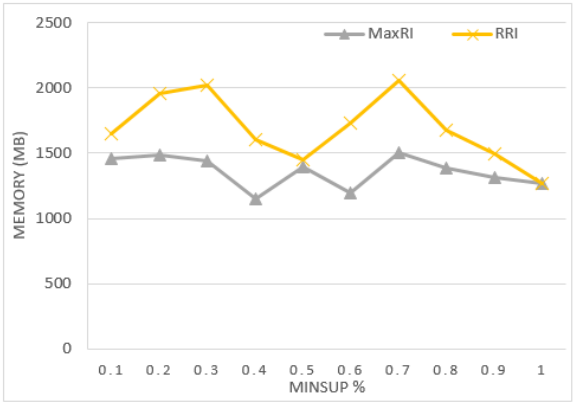

Figure 5: Memory cost for Accidents datasets.

6 CONCLUSION

The importance of rare itemset mining comes from the fact that discovering abnormal events (rare itemsets) as early as possible would avoid adverse outcomes. Traditional methods generate a vast amount of rare itemsets, which lead to extensive downstream analysis. Furthermore, the performance of the traditional methods degrades while mining rare itemsets from dense datasets.

In this paper, we proposed an approach for recovering the interesting rare itemsets. The proposed MaxRI algorithm efficiently retrieves a compact representation of rare itemsets without redundant ones. The experimental results show that extracting representative rare itemsets reduces runtime and memory cost. To recover the subset rare itemsets of the representative rare itemsets, we proposed the RRI algorithm to mine interesting subset rare itemsets of a specific length from representative rare itemsets generated by the MaxRI algorithm. For future work, MaxRI and RRI algorithms can be extended to deal with a massive amount of datasets by utilizing the MapReduce processing paradigm.

REFERENCES

[1] Agrawal, R., Srikant, R., Fast algorithms for mining association rules in large databases, Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pp. 487–499,1994.
[2] Chan, P. K., Stolfo, S. J., toward scalable learning with non-uniform class and cost distributions: a case study in credit card fraud detection. In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pages 164-168, 2001.
[3] Ji, Y., Ying, H., Tran, J., Dews, P., Mansour, A., and Massanari, R. M., A method for mining infrequent causal associations and its application in finding adverse drug reaction signal pairs, IEEE transactions on Knowledge and Data Engineering, pp.721-733, 2012.
[4] Weiss, G. M., and Hirsh, H., Learning to predict rare events in event sequences, In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, pp. 359-363, 1998.
[5] Koh, Y. S., and Rountree, N., Finding sporadic rules using apriori-inverse, In PacificAsia Conference on Knowledge Discovery and Data Mining (PAKDD), Springer, pp. 97-106, 2005.
[6] Szathmary, L., Napoli, A., and Valtchev, P., Towards rare itemset mining, In IEEE International Conference on Tools with Artificial Intelligence (ICTAI) 2007, IEEE, pp. 305–312, 2007.
[7] Weiss G.M., Mining with rarity: a unifying framework, ACM SIGKDD Explorations Newsletter, pp. 7–19, 2004.
[8] Kataria, M., Oswald, C., and Sivaselvan, B. , A Novel Rare Itemset Mining Algorithm Based on Recursive Elimination, Advances in Intelligent Systems and Computing, In Software Engineering, pp. 221-233, 2019.
[9] Tsang, S., Koh, Y. S., and Dobbie, G. , RP-Tree: Rare Pattern Tree Mining, In Data Warehousing and Knowledge Discovery (DaWaK), Lecture Notes in Computer Science, Springer, pp. 277-288, 2011.
[10] Darrab, S., Broneske, D., and Saake, G., RPP algorithm: a method for discovering interesting rare itemsets, in The the International Conference on Data Mining and Big Data, Springer-Nature in Communications in Computer and Information Science (CCIS), Springer, 2020.
[11] Darrab, S.and ERGENÇ, B., Frequent pattern mining under multiple support thresholds, the International Conference on Applied Computer Science (ACS), Wseas Transactions on Computer Research, pp. 1-10, 2016.
[12] Adda, M., Wu, L.,and Feng, Y., Rare itemset mining, In Sixth International Conference on Machine Learning and Applications (ICMLA), IEEE, pp. 73-80, 2007.
[13] Troiano, L., Scibelli, G., and Birtolo, C., A fast algorithm for mining rare itemsets, In Proceedings of the International Conference on Intelligent Systems Design and Applications, IEEE Computer Society Press, pp. 1149-1155, 2009.
[14] Liu, B., Hsu, W., and Ma, Y. , Mining association rules with multiple minimum supports, In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 337-341, 1999.
[15] Kiran, R. U., and Re, P. K, An improved multiple minimum support based approach to mine rare association rules, In Computational Intelligence and Data Mining (CIDM), IEEE, pp. 340-347, 2009.
[16] Han, J., Pei, J., and Yin, Y., Mining frequent patterns without candidate generation, In Proceedings of the International Conference on Management of Data (ACM SIGMOD), PP. 1-12, 2000.
[17] Darrab, S., and Ergenc, B., Vertical pattern mining algorithm for multiple support thresholds, International Conference on Knowledge Based and Intelligent Information and Engineering (KES), Procedia Computer Science 112, pp. 417-426, 2017.
[18] Hu, Y. H., and Chen, Y. L., Mining association rules with multiple minimum supports: a new mining algorithm and a support tuning mechanism, Decision Support Systems, pp. 1-24, 2006.
[19] Kiran, R. U., and Reddy, P. K., Novel techniques to reduce search space in multiple minimum supports-based frequent pattern mining algorithms, In Proceedings of the international conference on extending database technology (EDBT), pp. 11-20, 2011.
[20] Vo, B., Pham, S., Le, T., and Deng, Z. H. , A novel approach for mining maximal frequent patterns, Expert Systems with Applications, pp. 178-186, 2017
[21] Lazaar, N., Lebbah, Y., Loudni, S., Maamar, M., Lemière, V., Bessière, C.,and Boizumault, P., A global constraint for closed frequent pattern mining, In International Conference on Principles and Practice of Constraint Programming, Springer, pp. 333-349, 2016.

[22] Darrab S., Broneske D., and Saake G., Modern Application and Challenges for Rare Itemset Mining, in the International Conference on Knowledge Discovery, International Journal of Machine Learning and Computing (IJMLC), 2019.

[23] Borah, A., and Nath, B., Rare pattern mining: challenges and future perspectives, Complex and Intelligent Systems, pp. 1-23, 2019.

[24] Frequent Itemset Mining Dataset Repository, http://fimi.uantwerpen.be/data/.