Applied Intelligence (2022) 52:3387–3402
https://doi.org/10.1007/s10489-021-02574-1

FRI-miner: fuzzy rare itemset mining

Yanling Cui[1] · Wensheng Gan[2,3] · Hong Lin[4] · Weimin Zheng[1]

Accepted: 22 March 2021 / Published online: 6 July 2021

Abstract
Data mining is a widely used technology for various real-life applications of data analytics and is important to discover valuable association rules in transaction databases. Interesting itemset mining plays an important role in many real-life applications, such as market, e-commerce, finance, and medical treatment. To date, various data mining algorithms based on frequent patterns have been widely studied, but there are a few algorithms that focus on mining infrequent or rare patterns. In some cases, infrequent or rare itemsets and rare association rules also play an important role in real-life applications. In this paper, we introduce a novel fuzzy-based rare itemset mining algorithm called FRI-Miner, which discovers valuable and interesting fuzzy rare itemsets in a quantitative database by applying fuzzy theory with linguistic meaning. Additionally, FRI-Miner utilizes the fuzzy-list structure to store important information and applies several pruning strategies to reduce the search space. The experimental results show that the proposed FRI-Miner algorithm can discover fewer and more interesting itemsets by considering the quantitative value in reality. Moreover, it significantly outperforms state-of-the-art algorithms in terms of effectiveness (w.r.t. different types of derived patterns) and efficiency (w.r.t. running time and memory usage).

Keywords Quantitative data · Fuzzy-set theory · Rare pattern · Fuzzy data mining

1 Introduction

Association mining [1–3] of a transaction database is performed to determine association rules between a set of itemsets, for example, a set of events containing {*diaper*, *beer*} and {*milk*, *bread*}, which are often analyzed in market basket analysis. Several useful and interesting phenomena can be explored based on the association rules. By discovering the connection between items/objects, we can create a suitable market plan, form a suitable marketing strategy [4–6], and effectively apply the data to all types of analysis. Different data-mining methods are used depending on the requirements of various applications. The examples of such methods include frequent pattern mining (FPM) [1, 3, 7], utility-driven pattern mining [8–11], sequential pattern mining [12–14], and rare pattern mining (RPM) [15–18]. FPM is commonly adopted to extract association rules from a transaction database. In general, association rules are categorized as "frequent" or "rare" according to the specified minimum support threshold (*minSup*). "Frequent" refers to common or anticipated phenomena, while "rare" represents infrequent or previously unknown phenomena; thus, varied information can be extracted from the database.

In real life, it is possible to buy multiple copies of the same item in a transaction database, and mining association rules from such a quantitative database is an important task. The fuzzy set theory, which was first proposed by Zadeh in 1965 [19], is more suitable for dealing with quantitative values and expressing appropriate language values because it can help people better understand knowledge. Each element

✉ Wensheng Gan
wsgan001@gmail.com

✉ Weimin Zheng
zhengweimin@sdust.edu.cn

Yanling Cui
ylcui001@gmail.com

Hong Lin
lhed9eh0g@gmail.com

1 College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, Shandong, China

2 College of Cyber Security, Jinan University, Guangzhou 510632, Guangdong, China

3 Guangdong Artificial Intelligence and Digital Economy Laboratory (Pazhou Lab), Guangdong, China

4 Department of Computer Sciences, Guangdong University of Technology, Guangzhou 510006, China

in the fuzzy set can be assigned a membership degree to indicate the degree to which it belongs, such as {*A.low*}, {*B.mid*}. Chan and Au [20] first proposed an Apriori-like [1], namely F-APACS, to discover fuzzy association rules. Kuok et al. [21] proposed a method for processing quantitative data. Hong et al. [22] proposed an FDTA algorithm to process a quantitative database. In contrast to these Apriori-like algorithms, several fuzzy pattern mining algorithms based on tree structures were proposed. Lin et al. [23] proposed a fuzzy frequent pattern tree that can effectively discover fuzzy frequent itemsets (FFIs). Lin et al. [24] proposed a compressed FFP (CFFP)-tree algorithm. Although this CFFP-tree-based algorithm can reduce the number of tree nodes by using an additional array on each node, it is requires computational expenses to save the array. In addition, if the transactions are large, the spatial complexity of each node is high. Therefore, Lin et al. [25] proposed the UBFFP-tree algorithm to solve the problem of CFFP-tree overhead, which uses the same global sort strategy as in CFFP-tree to construct trees. Each term in the transaction is obfuscated by retaining only the language terms with the largest cardinality later in the process. Subsequently, Lin et al. [26] proposed the FFI-Miner algorithm to discover a complete set of FFIs without generating candidates. By adopting the fuzzy-list structure, the necessary information is preserved in the mining process. They also proposed an effective pruning strategy that can reduce the search space and further accelerate the mining process. Some effective algorithms for mining FFIs for association rules are still being studied.

Fuzzy-based data mining is simple and similar to human reasoning. According to the algorithms mentioned above, a fuzzy theory has been extensively developed [19] for application to FPM. However, the rare pattern mining (RPM) remains to be explored further. FPM, which is relatively mature, can find often-appearing or expected phenomena. In contrast, RPM can usually discover unknown or unexpected phenomena. Based on the existing algorithms, it is known that RPM can discover phenomena that are important in real life. For example, we assume that A and B represent two symptoms, and C represents a disease. In general, symptom A may lead to disease C. After rare pattern mining, we may find that symptom B also leads to disease C, which will be of great help to the medical industry. The performance of students with poor scores can also be determined, and their learning conditions can be adjusted accordingly. Apriori-like algorithms were initially used to determine frequent or rare association rules. If the minimum support (*minSup*) threshold is set extremely small, explosive growth occurs. However, if it is set extremely large, some useful association rules may be ignored. Based on the above problems, we attempt to use two thresholds, the minimum rare support (*minRSup*) and the minimum frequent support (*minFSup*), to achieve a better effect. As mentioned before, a quantitative transaction database can be processed with linguistic meaning using fuzzy theory. In the FFI-Miner algorithm [26], which aims to mine FFIs, while fuzzy rare but quite interesting itemsets are ignored.

To the best of our knowledge, RPM based on fuzzy theory has not yet been studied. To this end, a novel algorithm named FRI-Miner for mining fuzzy rare itemsets (FRIs) from a quantitative transaction database is proposed in this paper. The major contributions of this study are as follows.

1. This study is the first to formulate the problem of fuzzy RPM with linguistic meaning. The proposed fuzzy-list-based FRI-Miner algorithm addresses this problem successfully. As a fuzzy-theoretic data-driven model, it is explainable and similar to human reasoning that is more useful for decision making.
2. Fuzzy rare itemsets in FRI-Miner are categorized into three types: (1) containing only fuzzy rare items; (2) containing any combination of fuzzy rare items and fuzzy frequent items; and (3) containing only fuzzy frequent items. The first type is easy to understand; in the third type, fuzzy frequent itemsets that are themselves frequent may actually be fuzzy rare itemsets.
3. Several pruning strategies that utilize the properties of fuzziness, rare pattern, and fuzzy support are designed to successfully reduce the search space of FRI-Miner.
4. Experiments on several benchmark databases are conducted to show that the proposed FRI-Miner algorithm has better effectiveness and mining efficiency compared to those of existing methods.

The remainder of this paper is organized as follows: In Section 2, we review previous related work in the field of FPM and RPM. In Section 3, we present the definition and basic concepts of rare itemsets and fuzzy theory and then formulate the problem of FRI mining. The details of the proposed FRI-Miner algorithm are presented in Section 4. In addition, we use an example to illustrate the details of FRI-Miner for ease of understanding. The experimental evaluation is provided in detail in Section 5. Finally, the conclusions and future work are presented in Section 6.

2 Literature review

In this section, we briefly discuss fuzzy pattern mining and RPM. Then, we highlight the importance of fuzzy RPM and further discuss several applications.

2.1 Fuzzy pattern mining

According to the existing FPM algorithms, the data types can be roughly categorized as Boolean, ordered, or quantitative

[2, 3, 7]. Traditional FPM aims at processing data that do not contain quantitative values, although quantitative data are common in real life. To solve the processing problem of a quantitative database, the quantitative value of a project can be converted into a language term with a fuzzy degree by using fuzzy set theory [19]. Meaningful fuzzy association rules can be discovered in real life. The addition of fuzzy concepts fuzzifies the quantitative values. This is also convenient to better understand the true meaning of the data and make the data simpler and easier to understand. For example, if the database does not contain the quantity value corresponding to each item, then the frequency of the item is only related to the Boolean value. However, for databases containing quantitative values, the frequency of occurrence of each item is determined by the quantitative values of the items.

Frequent itemset mining based on fuzzy theory was further elaborated by the Apriori algorithm [1]. Fuzzy association rules are induced by the hierarchical intelligent mining of the FFI. First, according to the predefined membership function, the quantization value of the item is converted into a language term, which not only reflects the occurrence frequency of the item in the database, but also reflects the support of the itemset. Therefore, several algorithms for mining fuzzy itemsets that meet the minimum support threshold through deterministic factors have been designed. The F-APACS algorithm for mining fuzzy association rules was first proposed by Chan and Au [20]. Kuok et al. [21] proposed an algorithm by processing quantitative attributes to discover fuzzy association rules, and Hong et al. [22, 27] proposed the FDTA algorithm to process quantitative databases using fuzzy set theory. Subsequently, a fuzzy frequent pattern tree [28] was proposed. Several tree-based algorithms for mining FFIs have been proposed [23, 25], which are based on the FP-growth method [29] for mining frequent itemsets. These algorithms first construct a tree structure of fuzzy frequent items through the reserved frequent itemsets and then further extract more fuzzy frequent patterns from the constructed tree structure. However, the mining process with a pattern-growth mechanism [29] is often complicated and requires the storage of numerous tree nodes with additional information. To summarize, all these fuzzy-based FPM algorithms can only discover expected frequent phenomena.

2.2 Rare pattern mining

Based on the Apriori algorithm [1], many frequency-based data mining algorithms have been extended, but the mining results mostly correspond to common or expected phenomena. To address this, another data mining framework, rare pattern mining (RPM) [16, 30–32], has been introduced, and many algorithms for RPM have been proposed in recent years. In contrast to frequent and common patterns, discovering rare patterns may be more useful in some cases (e.g., itemsets and association rules), which are important for real-life applications.

Most of static rare itemset mining algorithms can be roughly divided into support threshold, no support threshold, and constraints [16]. Because we use the support threshold in this study, we provide a brief overview of RPM with the support threshold. Because the support threshold for rare itemsets is lower than that for frequent itemsets, the generation of rare itemsets is better realized by setting lower or distinct support thresholds. To address the "*rare item problem*" in FPM, in 1999, Liu et al. [33] proposed the MSApriori algorithm that adopted multiple minimum supports (MMS) to successfully discover rare itemsets. In contrast to this Apriori-like method, several set-enumeration tree based methods were proposed for RPM, such as the frequency-driven FP-ME algorithm [30] and the utility-driven HUIM-MMU [34] and HIMU [35] algorithms. The Apriori-reverse algorithm proposed by Koh and Rountree [36] aims to discover rare rules with respect to completely dispersed itemsets, which only contain items below the maximum support threshold. Later, a rare itemset mining algorithm (ARIMA) was proposed by Szathmary et al. [17]. Troiano et al. [37] introduced the rarity algorithm, which is a top-down rare itemset mining algorithm. Up to now, a number of RPM algorithms have been extensively proposed, such as CFP-growth++ [15]. Among them, several studies are designed to deal with dynamic data streams [38–40]. There are numerous candidates for most algorithms based on Apriori mechanism. The RP-tree-based algorithm [18] discovers rare patterns that meet the conditions between *minRSup* and *minFSup*.

2.3 Applications of fuzzy rare pattern mining

For some real applications, unusual rare patterns are more important and useful than frequent ones. In some application domains, RPM is more suitable for intelligent systems. The result of network intrusion can be obtained by detecting whether the network is abnormal. In medicine, sudden changes are diagnosed by finding data that are different from those corresponding to normal health. In an insurance company, by finding rare people who need high-risk claims, making reasonable marketing strategies, and so on. In recent years, the field of RPM [16] has been further developed. The emergence of the RPM has also made a significant contribution to the research community. However, these previous RPM algorithms rarely involve fuzzy theory. In contrast, many FPM algorithms based on fuzzy theory have been widely used in the mining of quantitative data, while RPM dealing with quantitative data is extremely rare. To the best

of our knowledge, RPM dealing with Boolean or quantitative data based on fuzzy theory has not yet been studied. To this end, we propose an effective algorithm that uses fuzzy theory to discover the interesting rare patterns from a quantitative transaction database.

3 Preliminaries and problem formulation

In this section, we introduce some basic concepts, principles of fuzzy-driven pattern mining and RPM. Some definitions in previous research are adopted here to present common concepts clearly. Further details regarding the background of the fuzzy FPM can be found in Ref. [23, 26].

3.1 Preliminaries

$I = \{i_1, i_2, \ldots, i_m\}$ represents a finite set I composed of m different items. $D = \{t_1, t_2, \ldots, t_n\}$ represents a transactional database D ($1 \leq q \leq n$) composed of n different items, in which each transaction t_q is a subset of I. Each transaction contains a unique identifier *tid*, and each item lists the number of quantities as v_{iq}. If an itemset consists of k different items, we call it k-itemset, and if each item in the k-itemsets is included in the transaction t_q, we call the itemset a subset of the transaction t_q. The minimum support is defined as *minRSup*, and the maximum support is defined as *minFSup*. The specified member function is set to μ, which can be adjusted according to the user's needs. In this study, all items in our database are represented using a single fuzzy membership function, as shown in Fig. 1. We set three language terms in the adopted membership functions: *low*, *middle*, and *high*. The quantitative value of each item is obscured by the same member function as the corresponding language term.

In this paper, we present a running example using a quantitative transactional database, as shown in Table 1. There are six items, (A), (B), (C), (D), (E), (F), and eight transactions $\{t_1, t_2, \ldots, t_8\}$. We set the minimum support threshold to *minRSup* (= 25%) and the maximum support threshold to *minFSup* (= 50%). The idea is to filter out items that seem insensible and similar to noise; thus, this can help to reduce unnecessary search space.

Definition 1 (The attributes of a quantitative database) They are represented by the language variable R_i, and the fuzzy language terms are represented by the natural language as $(R_{i1}, R_{i2}, \ldots, R_{il})$. R_i can be defined by the membership function μ.

For example, in this study, the membership function μ applied in the running example is shown in Fig. 1. There are five items in Table 1: (A), (B), (C), (D), (E), and (F). The three language terms are expressed as $Low(L)$, $Middle(M)$, and $High(H)$. In transaction t_1, item B is denoted by $(B.L)$, $(B.M)$, and $(B.H)$, as shown in Fig. 1. The other items in these transactions are calculated similarly to those of item B.

Definition 2 (The quantitative value of an item) In the quantitative database, the value of an item is expressed as v_{iq}, which represents the number of this item i in transaction t_q.

For example, in transaction t_2, the quantitative values of the items as (B) and (D) are v_{B2} (= 8) and v_{D2} (= 3), respectively. Here, we can clearly see what it means.

Definition 3 (Fuzzy set) In the fuzzy stage, a fuzzy set refers to the set of fuzzy language terms with a membership degree (fuzzy value) converted from the quantitative value v_{iq} of the item i in the transaction database t_q by the membership function μ. Its specific expression is as follows:

$$f_{iq} = \mu_i(v_{iq}) = \left(\frac{fv_{iq1}}{R_{i1}} + \frac{fv_{iq2}}{R_{i2}} + \cdots + \frac{fv_{iqh}}{R_{il}} \right), \quad (1)$$

where l refers to the number of fuzzy language terms in which the membership function μ converts to I, R_{il} represents the l-th fuzzy language term of item i, and fv_{iql} represents the membership of item i in the quantitative value v_{iq} (fuzzy value) of the l-th fuzzy language term R_{il}'s, where $fv_{iql} \subseteq [0, 1]$.

For example, in transaction t_2, we convert the quantitative value of the items in Table 1 to the membership degree using the membership function. Here, we provide a detailed description of the quantitative value of (B) in Fig. 1 as $\left(\frac{0.6}{B.L} + \frac{0.4}{B.M}\right)$. The transformation of all other transactions is similar to the transformation of item (B) in transaction t_2. The specific results are listed in Table 2.

Definition 4 (Membership degree) The support after the membership function is converted to membership degree and expressed as $sup(R_{il})$. The sum of the scalar cardinality of the fuzzy value of R_{il} is expressed as:

$$sup(R_{il}) = \sum_{R_{il} \subseteq t_q \wedge t_q \in Q'} fv_{\cdot} \quad (2)$$

where database Q' is the database transformed by the membership function μ, which is the same as the original database D.

For example, the support of the fuzzy terms $(B.M)$ appears in transactions $t_1, t_2, t_3, t_4, t_5, t_6, t_7$, and t_8, as shown in Table 2. Thus, its support in the running database can be

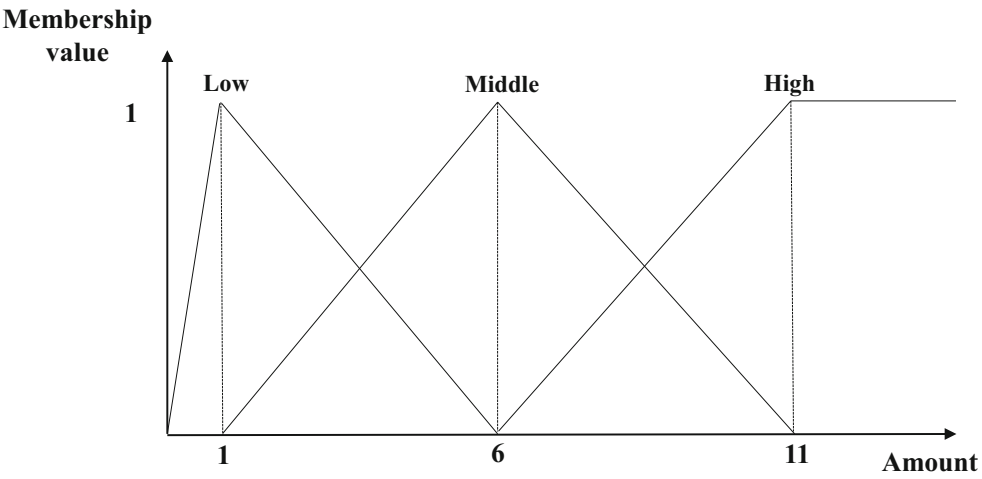


Fig. 1 The used linear membership functions of linguistic 3-terms

calculated as $sup(B.M) = 0.8 + 0.6 + 0.6 + 0.8 + 0.8 + 0.8 + 0.6 + 0.8 = 5.8$.

Definition 5 (The minimum fuzzy value of an *k*-itemset) An itemset X is composed of k-items ($k \geq 1$), and its support degree is expressed as $sup(X)$, which represents the sum of the minimum fuzzy values of k-items contained in X. The definition is as follows:

$$sup(X) = \{X \in R_{il} | \sum_{X \subseteq t_q \wedge t_q \in Q'} min(fv_{i_j ql}, fv_{i_m ql}), \\ i_j, i_m \in X, i_j \notin i_m\}, \quad (3)$$

where the items in X do not intersect each other.

For example, the fuzzy 2-itemset $(A.L, D.H)$ appears in transactions t_1, t_3, t_6, and t_7, as shown in Table 2. Thus, the support of $(A.L, D.H)$ can be calculated as $sup(A.L, D.H) = \{min(0.6, 0.8) + min(0.6, 0.6) + min(0.8, 0.2) + min(0.8, 0.6)\} = \{0.6 + 0.6 + 0.2 + 0.6\} = 2.0$. Note that here the fuzzy-based support is different from the support concept of FPM, and the later equals to the number of transactions contains X.

In a transaction database, the mining of some common itemsets aims at discovering frequent itemsets. In previous methods, a minimum frequent support *minFSup* is generally set, and itemsets that meet the *minFSup* specified by the user are preserved. Thus, more frequent itemsets that meet these criteria are discovered. For RPM, we set the minimum rare support *minRSup*. RPM discovers itemsets that meet the criteria of *minRSup*. However, the support of these rare itemsets cannot be greater than that of *minFSup*. For example, we assume that *sup(A)* = 1.0, and *sup(B)* = 2.5. If we set the *minRSup* to 2.0, and the *minFSup* to 3.0, then, we find that item A does not meet the *minRSup*. The support of item B is greater than that of *minRSup* and less than *minFSup*. Thus, item B is a rare pattern that we intend to discover.

Definition 6 (Fuzzy rare itemset) Considering quantitative data, the user specifies the specific minimum rare support threshold and minimum frequent support threshold, which are *minRSup* and *minFSup*, respectively. Only the itemsets that satisfy two conditions can be considered as fuzzy rare itemsets (FRIs), as follows:

$$FRIs \leftarrow \{X | minRSup \times |D| \leq sup(X) \\ \leq minFSup \times |D|\}. \quad (4)$$

Table 1 A quantitative database

tid	Transaction
t_1	A:3, B:5, D:10, E:9
t_2	B:8, D:3
t_3	A:3, B:8, D:9, F:5
t_4	B:5, C:4, D:11, E:2
t_5	B:7, C:3, D:5, F:3
t_6	A:2, B:5, C:3, D:7
t_7	A:2, B:4, D:9, F:2
t_8	B:5, C:2, D:10, E:3

Table 2 Transformed results from Table 1

tid	Fuzzy transaction
t_1	$\frac{0.6}{A.L} + \frac{0.4}{A.M}, \frac{0.2}{B.L} + \frac{0.8}{B.M}, \frac{0.2}{D.M} + \frac{0.8}{D.H}, \frac{0.4}{E.M} + \frac{0.6}{E.H}$
t_2	$\frac{0.6}{B.M} + \frac{0.4}{B.H}, \frac{0.6}{D.L} + \frac{0.4}{D.M}$
t_3	$\frac{0.6}{A.L} + \frac{0.4}{A.M}, \frac{0.6}{B.M} + \frac{0.4}{B.H}, \frac{0.4}{D.M} + \frac{0.6}{D.H}, \frac{0.2}{F.L} + \frac{0.8}{F.M}$
t_4	$\frac{0.2}{B.L} + \frac{0.8}{B.M}, \frac{0.4}{C.L} + \frac{0.6}{C.M}, \frac{0}{D.M} + \frac{1}{D.H}, \frac{0.8}{E.L} + \frac{0.2}{E.M}$
t_5	$\frac{0.8}{B.M} + \frac{0.2}{B.H}, \frac{0.6}{C.L} + \frac{0.4}{C.M}, \frac{0.2}{D.L} + \frac{0.8}{D.M}, \frac{0.6}{F.L} + \frac{0.4}{F.M}$
t_6	$\frac{0.8}{A.L} + \frac{0.2}{A.M}, \frac{0.2}{B.L} + \frac{0.8}{B.M}, \frac{0.6}{C.L} + \frac{0.4}{C.M}, \frac{0.8}{D.M} + \frac{0.2}{D.H}$
t_7	$\frac{0.8}{A.L} + \frac{0.2}{A.M}, \frac{0.4}{B.L} + \frac{0.6}{B.M}, \frac{0.4}{D.M} + \frac{0.6}{D.H}, \frac{0.8}{F.L} + \frac{0.2}{F.M}$
t_8	$\frac{0.2}{B.L} + \frac{0.8}{B.M}, \frac{0.8}{C.L} + \frac{0.2}{C.M}, \frac{0.2}{D.M} + \frac{0.8}{D.H}, \frac{0.6}{E.L} + \frac{0.4}{E.M}$

We take an example of 1-itemset. Through the calculation of the fuzzy values, we find that $sup(A.L) = 2.8$, $sup(B.M) = 5.8$, $sup(C.L) = 2.4$, and $sup(D.H) = 4.0$. According to the definition of FRIs, we find that $(A.L)$ and $(C.L)$ are rare itemsets, whereas $(B.M)$ and $(D.H)$ are frequent itemsets.

3.2 Problem statement

To date, many algorithms have been developed for mining frequent patterns in quantitative databases. Corresponding with the emergence of quantitative values, fuzzy theory has also merged. Therefore, a mining algorithm for fuzzy frequent patterns is proposed accordingly. However, there are many interesting but rare patterns that are ignored, mostly because of setting a low support.

In this study, the problem of fuzzy rare pattern mining (fuzzy RPM for short) is formulated as follows. Given a quantitative database, the specific minimum support threshold and maximum support threshold, which are *minRSup* and *minFSup*, respectively. The goal of fuzzy RPM is to discover the complete set of FRIs that satisfies two conditions.

4 Proposed fuzzy mining algorithm: FRI-miner

For the mining of quantitative databases, previous studies have shown that fuzzy theory can play an important role. Through many previous statements, we find that it has been relatively mature in the mining of frequent itemsets, while RPM based on fuzzy theory has not yet been studied. In this study, we apply fuzzy theory to process the quantitative database, obtain the candidates, and finally discover FRIs by using the fuzzy-list structure [26]. In summary, the specific steps of the FRI-Miner algorithm include: 1) fuzzification phase, 2) construction of fuzzy-list, and 3) recursively mining FRIs, which are presented as follows.

4.1 Fuzzification phase

FRI-Miner first fuzzes the quantitative database and converts v_{iq} in quantitative data into the membership degree of the corresponding fuzzy terms through a membership function. Next, it forms a new fuzzy database by using the maximum scalar cardinality and support-ascending order.

Definition 7 (Maximum scalar cardinality) For an item i, it uses the corresponding fuzzy term R_{il} to denote its corresponding quantitative value v_{iq}. In the expressed fuzzy terms, we find the fuzzy terms that should be reserved according to the maximum scalar cardinality. This can represent the corresponding language variables as item i.

For example, in Table 2, the transformed fuzzy terms with their summed fuzzy values of the linguistic variable (A) are $(A.L: 2.8, A.M: 1.2, A.H: 0)$. Thus, the fuzzy term $(A.L)$ is used to represent the linguistic variable of (A), which can be used for the later mining process of FRIs.

Based on the reserved R_{il} of the maximum scalar cardinality, the fuzzy quantitative database was modified to form a new fuzzy-based database. The revised database from Table 2 is presented in Table 3. Fuzzy items reserved by each transaction t_q are then sorted into ordered fuzzy items according to the support-ascending order. Note that the items retained here meet the minimum level of support. In addition to rare items, these items also have frequent items. The cross combination between them produces a new itemset that meets the two conditions.

Definition 8 (The support-ascending order) In the fuzzy transaction database, fuzzy items that satisfy the fuzzy value condition are retained according to the maximum scalar cardinality in the transaction t_q. Based on the fuzzy values of the reserved fuzzy items, they are sorted according to the support-ascending order to prepare for the subsequent computation.

For example, we can obtain the result through the transformed fuzzy database, based on the maximum scalar cardinality and the support-ascending order.

4.2 Fuzzy-list construction phase

In FRI-Miner, first, v_{iq} in a quantitative database is converted into a membership degree corresponding to the corresponding fuzzy terms through the membership function. Then, we form a novel fuzzy database and then construct the corresponding fuzzy-list structure [26]. We keep the fuzzy terms in L_1, which uses three fields that make up the fuzzy-list: the transaction identifier (tid), the internal fuzzy value (if), and the remaining fuzzy value (rf). We briefly introduce these details.

Table 3 A revised database

tid	Fuzzy transaction
t_1	$\frac{0.6}{A.L}, \frac{0.8}{D.H}, \frac{0.8}{B.M}$
t_2	$\frac{0.6}{B.M}$
t_3	$\frac{0.6}{A.L}, \frac{0.6}{D.H}, \frac{0.6}{B.M}$
t_4	$\frac{0.4}{C.L}, \frac{1.0}{D.H}, \frac{0.8}{B.M}$
t_5	$\frac{0.6}{C.L}, \frac{0.8}{B.M}$
t_6	$\frac{0.6}{C.L}, \frac{0.8}{A.L}, \frac{0.2}{D.H}, \frac{0.8}{B.M}$
t_7	$\frac{0.8}{A.L}, \frac{0.6}{D.H}, \frac{0.6}{B.M}$
t_8	$\frac{0.8}{C.L}, \frac{0.8}{D.H}, \frac{0.8}{B.M}$

Definition 9 (Transaction identifier) It represents a fuzzy term R_{il} in transaction t_q, which is a subset of the corresponding transaction in t_q is denoted as $R_{il} \subseteq t_q$. Here, we use the corresponding (tid) to represent the presence in that transaction.

According to the initial construction of the fuzzy-list, as shown earlier in L_1, the support ascending order is used to obtain $(C.L < A.L < D.H < B.M)$ in Fig. 2. For example, in Table 3 and Fig. 2, the fuzzy item $(B.M)$ exists in t_1, t_2, t_3, t_4, t_5, t_6, t_7, and t_8.

Definition 10 (Internal fuzzy value) The fuzzy value (if) of the fuzzy term R_{il} in the transaction t_q is denoted as $if(R_{il}, t_q)$.

For example, in Table 3, the internal fuzzy values of $(B.M)$ in t_1 and t_2 are $if(B.M, t_1) = 0.8$, and $if(B.M, t_2) = 0.6$, respectively.

Definition 11 (The resting fuzzy value [26]**)** In the fuzzy-list, the resting fuzzy value of R_{il} is expressed as $rf(R_{il}, t_q)$. It represents the maximum fuzzy value obtained by performing a union operation. In other words, the upper bound value of all fuzzy terms R_{il} in t_q, as shown in Fig. 2. It is defined as:

$$rf(R_{il}, t_q) = max\{if(z, t_q) | z \in (t_q / R_{il})\}. \tag{5}$$

According to Table 3, formed by the support-ascending order, the $rf(C.L, t_4)$ is calculated as $max\{0.8, 1.0\} = 1.0$, and $rf(C.L, t_5)$ is calculated as $max\{0.8\} = 0.8$, the $rf(C.L, t_6)$ is counted as $max\{0.8, 0.2, 0.8\} = 0.8$, and the $rf(C.L, t_8)$ is $max\{0.8, 0.8\} = 0.8$. Details of the fuzzy-list structure can be referred to Ref. [26].

In Fig. 2, we can observe that the element (4, 0.4, 1.0) in the constructed fuzzy-list structure of $(C.L)$ indicates (tid, if, rf), 4 represents the transaction t_4, 0.4 represents the internal fuzzy value is 0.8, and 1.0 represents the resting fuzzy value after $(C.L)$ is 1.0. The other information for the fuzzy item is similarly represented.

The fuzzy-list of the 1-itemset performs the intersection operation to form a new fuzzy-list structure of k-itemsets $(k \geq 2)$. In the recombination process, the ones with the same tid are combined. Except for the item that needs to be calculated, the internal fuzzy value of the transaction corresponding to all eligible fuzzy k-items $(k \geq 2)$ takes the minimum value of the merged item as the remaining fuzzy value. In Fig. 3, the combined results of fuzzy 2-itemsets are shown as $(C.L, A.L)$, $(C.L, D.H)$, $(C.L, B.M)$, and so on.

According to the similar fuzzy-list in Fig. 2, we obtain the corresponding values of the three columns, and the corresponding support can be obtained according to the values of the second and third columns. They are defined as follows:

Definition 12 In the fuzzy-list, we can calculate the sum of the inner fuzzy values of an itemset R_{il} in a quantitative database, and it is denoted as $SUM(R_{il}.if)$ [26] and is defined as follows:

$$SUM(R_{il}.if) = \sum_{R_{il} \subseteq t_q \wedge t_q \in Q'} if(R_{il}, t_q). \tag{6}$$

For example, in Fig. 2, the sum of the internal fuzzy values of $(C.L)$ in D is calculated as $(0.4 + 0.6 + 0.6 + 0.8) = 2.4$.

Definition 13 In the fuzzy-list, we calculate the sum of the resting fuzzy values of R_{il} in the quantitative database D is defined as $SUM(R_{il}.rf)$ [26]. Here, the rf value is calculated from the third column as follows:

$$SUM(R_{il}.rf) = \sum_{R_{il} \subseteq t_q \wedge t_q \in Q'} rf(R_{il}, t_q). \tag{7}$$

For example, in Fig. 2, the sum of the resting fuzzy values for $(C.L)$ in D is calculated as $(1.0 + 0.8 + 0.8 + 0.8) = 3.4$.

Fig. 2 The constructed fuzzy-list structures

C.L		
tid	***if***	***rf***
4	0.4	1.0
5	0.6	0.8
6	0.6	0.8
8	0.8	0.8

A.L		
tid	***if***	***rf***
1	0.6	0.8
3	0.6	0.6
6	0.8	0.8
7	0.8	0.6

D.H		
tid	***if***	***rf***
1	0.8	0.8
3	0.6	0.6
4	1.0	0.8
6	0.2	0.8
7	0.6	0.6
8	0.8	0.8

B.M		
tid	***if***	***rf***
1	0.8	0
2	0.6	0
3	0.6	0
4	0.8	0
5	0.8	0
6	0.8	0
7	0.6	0
8	0.8	0

(C.L, A.L)		
tid	if	rf
6	0.6	0.8

(C.L, D.H)		
tid	if	rf
4	0.4	0.8
6	0.2	0.8
8	0.8	0.8

(C.L, B.M)		
tid	if	rf
4	0.4	0
5	0.6	0
6	0.6	0
8	0.8	0

(A.L, D.H)		
tid	if	rf
1	0.6	0.8
3	0.6	0.6
6	0.2	0.8
7	0.6	0.6

(A.L, B.M)		
tid	if	rf
1	0.6	0
3	0.6	0
6	0.8	0
7	0.6	0

(D.H, B.M)		
tid	if	rf
1	0.8	0
3	0.6	0
4	0.8	0
6	0.2	0
7	0.6	0
8	0.8	0

Fig. 3 The fuzzy-lists of fuzzy 2-itemsets

The fuzzy value in the fuzzy-list was used to obtain the total support. Here, we use the support-ascending order to construct a set of fuzzy-lists. According to the measurement of the support of the corresponding item, we obtain the rare itemsets between the support tends to the minimum and the maximum specified support threshold, and the fuzzy itemsets with the support above the maximum support. We also obtain a set of items that we can intersect further in Fig. 3. Then, it sorts the results according to the support-ascending order and adopts the pruning strategies to prune the search space with respect to an enumerated tree [41]. Then, the 1-itemset is gradually expanded to 2-itemsets, 3-itemsets, and other combinations, which are shown in Fig. 4. Finally, a set of rare itemsets that satisfy support can be obtained.

In Fig. 4, we narrow the search space by pruning to remove items below the minimum threshold. The remaining items are combined to form a new k-itemset. The structure of the fuzzy-list of the k-itemset is similar to that of the 1-itemset. The specific description is as follows:

Theorem 1 *According to the concept of FRIs, for an itemset X in the fuzzy-list structure, if its SUM(X.if) is no less than the minimum fuzzy rare support and no more than the maximum fuzzy frequent support, it is seen as a FRI. If min(SUM(X.if), SUM(X.rf)) of X is no less than the minimum fuzzy rare support, new itemsets are required to be generated. If the sum of the resting fuzzy values of R_{il} is no less than the minimum fuzzy rare support ($minRSup \times |Q'|$), extensions of R_{il} may be a FRI. If the summation of the resting fuzzy values of R_{il} is smaller than the minimum fuzzy rare support, any extensions of R_{il} will neither be a FRI nor a FFI. Thus, there is no need to construct a new fuzzy-list structure for its extension.*

Proof For $\forall t_q \supseteq X'$, suppose fuzzy term R_{il} is denoted as X, and X' is the extension of X ($X \subset X' \subseteq t_q \Rightarrow X'.tids \subseteq X.tids$), thus $(X' - X) = (X'/X)$ and $(X'/X) \subseteq (t_q/X)$. Since $if(X', t_q) = min\{if(X, t_q), rf(R_{il}, t_q)\}$, it holds $if(X') = SUM(X.rf)$ [26]. Thus, with the definition of a FRI, both *minRSup* and *minFSup* are able to determine

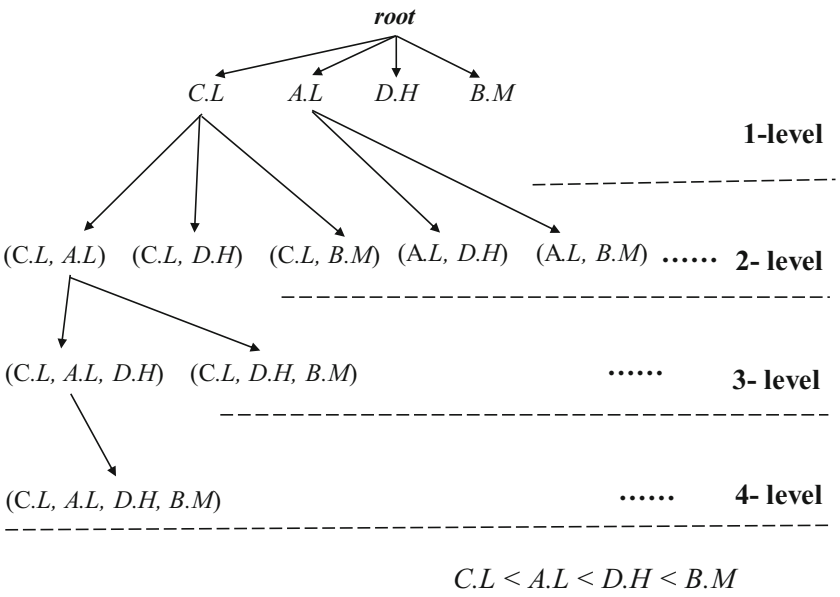


Fig. 4 An enumeration tree of the used example

the promising candidates for FRIs by quickly pruning the search space. □

For example, we can consider the 3-itemsets ($A.L$, $D.H$, $B.M$), which is an extension of 2-itemsets ($A.L$, $D.H$). Because the sum of the resting fuzzy values of ($A.L$, $D.H$) is calculated as (0.8 + 0 + 0.8 + 0.4) = 2.0. The extension ($A.L$, $D.H$, $B.M$) in the search space of FRI-Miner with respect to an enumeration tree [41] must be generated. For the 3-itemsets ($C.L$, $A.L$, $D.H$), which is the extension of 2-itemsets ($C.L$, $A.L$), the sum of the resting fuzzy values of ($C.L$, $A.L$) is calculated as $0.8 < 2.0$. Thus, the extension of ($C.L$, $A.L$, $D.H$) is unnecessary because the fuzzy value of the 3-itemsets is lower than the specified minimum fuzzy value. A detailed description of the FRI-Miner algorithm is presented in the following section. The pseudocode for the construction of the fuzzy-list here is similar to that of the frequency-based FFI-Miner algorithm [26].

4.3 Fuzzy rare itemset mining phase

Note that we have learned about both rare and frequent itemsets from the previous sections. In this section, we go into more details about the mining processes of FRIs. In general, there are three types of rare itemsets, as follows:

1. The itemsets with only fuzzy rare items: all items in an itemset are rare, and the itemset may be rare.
2. The itemsets with fuzzy rare and fuzzy frequent items: an itemset that contains rare and frequent items internally may be a rare itemset. For example, we can calculate the sum of the inner fuzzy values of itemset ($A.L$) is calculated as ($= 2.8 > 2.0$), and the sum of the inner fuzzy values of the itemset ($D.H$) is calculated as ($= 4.0 \geq 4.0$), but we can calculate the sum of the inner fuzzy values of the itemset ($A.L$, $D.H$) is calculated as ($= 2.0 \geq 2.0$).
3. The itemsets with only fuzzy frequent items: an itemset in which all items are frequent may be rare. For example, we can calculate the sum of the inner fuzzy values of the itemset ($D.H$) as 4.0. The sum of the inner fuzzy values of the itemset ($B.M$) is calculated as ($= 5.8 > 4.0$), but the value of the itemset ($D.H$, $B.M$) is calculated as ($4.0 > 3.8 > 2.0$).

In the above discussion, we have carried out detailed concepts, data structure, theorem, and strategies of the proposed FRI-Miner algorithm. Algorithm 1 describes the complete details of FRI-Miner for discovering FRIs. Notice that the itemset that satisfies $SUM(X.if) \geq minRSup \times |D|$ will be rare or frequent (lines 2-3), and it is used to generate the new itemsets (extensions of X, lines 11-12). This guarantees the correctness and completeness of the final discovered results of FRI-Miner.

Algorithm 1 FRI-miner.

Input: A quantitative database D; $minRSup$; $minFSup$.
Output: $FRIs$: all valid fuzzy rare itemsets.

1: First, it fuzzes the quantitative database to form a new fuzzy database, and then forms a corresponding fuzzy-list structure for each item.
2: **for** each fuzzy-list $X \in FLs$ **do**
3: **if** $SUM(X.if) \geq minRSup \times |D|$ **then**
4: $FLs \leftarrow X \cup FLs$; // the candidates
5: **if** $SUM(X.if) \leq minFSup \times |D|$ **then**
6: $FRIs \leftarrow X \cup FRIs$; // output the results
7: **end if**
8: **end if**
9: **if** $SUM(X.rf) \geq minRSup \times |D|$ **then**
10: initialize $exFLs \leftarrow null$;
11: **for** each fuzzy-list Y after $X \in FLs$ **do**
12: call the construct(X, Y) procedure, then put the built fuzzy-list into $exFLs$;
13: **end for**
14: call FRI-Miner ($exFLs$, $minRSup$);
15: **end if**
16: **end for**
17: return $FRIs$

According to the above pseudo-code, it can be known that the quantitative database as follows shown in Table 1 is converted to a fuzzy database as follows shown in Table 2 at the beginning of the algorithm, and the membership function μ is used to blur the quantitative value at this phase. The fuzzy items in the fuzzy database obtained in the first phase are clearly planned in the fuzzy-list structure, and the fuzzy value that meets the user-specified threshold is selected by the maximum scalar cardinality and support-ascending order strategy. It keeps the rare and frequent items that meet the conditions and outputs rare items. It combines the remaining fuzzy items in a tree structure and deletes the itemsets that do not meet the conditions through the pruning strategy. Finally, it outputs complete FRIs that meet the conditions.

5 Experimental evaluation

In this study, we address the problem of mining FRIs in a quantitative database. To the best of our knowledge, FRI-Miner is the first fuzzy-based algorithm for mining rare itemsets with linguistic meanings. It can discover rare itemsets that satisfy the two conditions of fuzzy values. The RP-Growth algorithm [18], which aims to mine rare itemsets, was selected as the baseline to evaluate the validity and performance of the proposed FRI-Miner algorithm.

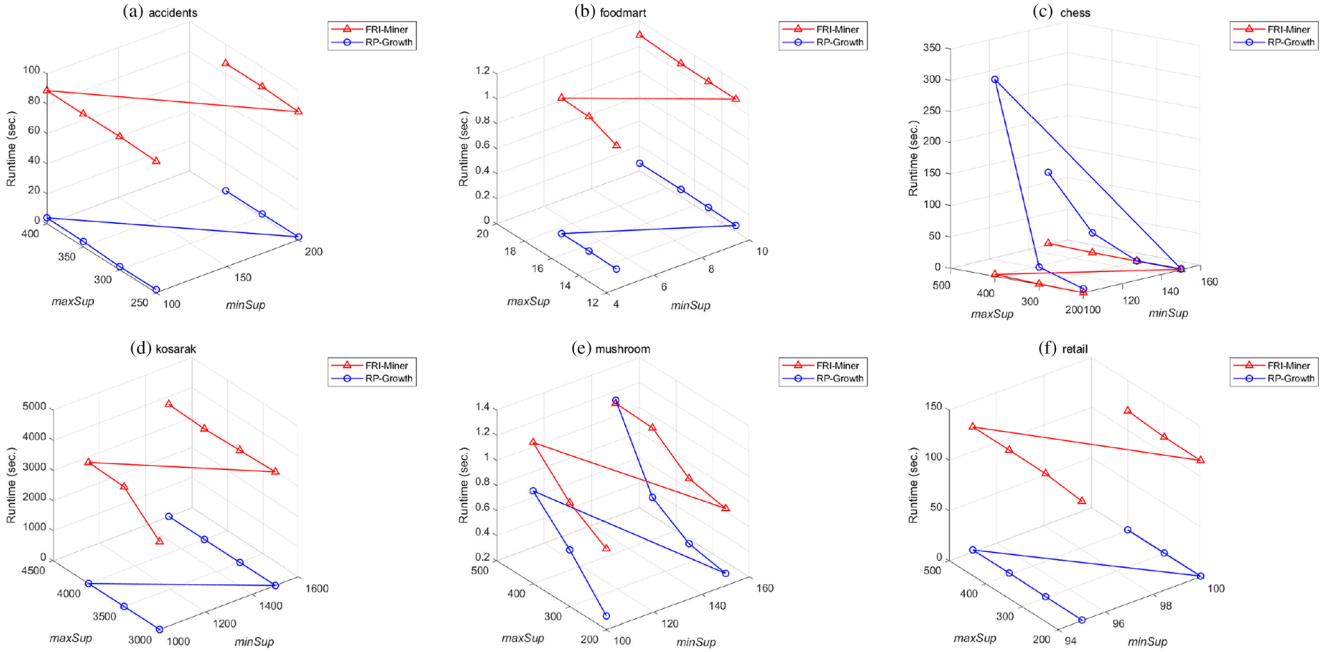


Fig. 5 Runtime vs. *minSup*/*maxSup*

A total of six benchmark datasets were used in this experiment: retail, accidents, foodmart, chess, mushroom, and kosarak. Real-life datasets are available in a public repository.[1] Their characteristics and descriptions can be found in previous studies [7, 42]. All algorithms were implemented using Java language and executed on an Intel Core 6 Duo 3.00 GHz machine with 8 GB of RAM running Windows 10.

We transform the quantitative datasets according to a predefined membership function. For the foodmart dataset, we used a different membership function, which is [100, 600, 1100]. The reason is to ensure that the data in the dataset are evenly distributed in the set membership function, so that the final result obtained is more accurate. The other datasets were distributed between the specified [1, 21, 31]. According to the previous algorithm, to evaluate the performance of the compared algorithms, we analyzed them from three aspects: running time, memory usage, and the number of generated patterns. For example, the changes in runtime can reflect whether the designed data mining algorithm is acceptable within a reasonable execution time. The results obtained with the compared algorithms are shown in Figs. 5, 7, Table 4, and Fig. 8, respectively. The detailed results are as follows.

5.1 Effectiveness

As mentioned before, FPM and RPM are two different mining tasks. Moreover, the fuzzy-theoretic-based RPM is also different from the traditional RPM. In the mining rare itemsets, two thresholds *minRSup* and *minFSup* are set in FRI-Miner. As shown in Table 4 and Fig. 5, the constraints of the two thresholds further enable the FRI-Miner algorithm to run efficiently, shorten the running time, and identify meaningful rare itemsets. First, FRI-Miner uses *minRSup* to remove itemsets that are lower than *minRSup*; that is, these itemsets without practical significance. Then, fuzzy-based rare itemsets are discovered from the potential candidates. In the experiments, *minRSup* and *minFSup* are expressed as *minSup* and *maxSup*, respectively, as shown in Table 4. Note that the number of *#FRIs* is derived by FRI-Miner, and the number of *#RIs* is derived by the RP-Growth.

In general, in the experimental process, the running time reflects the execution efficiency of the algorithm. According to the experimental results, we find that the running time of the FRI-Miner algorithm is relatively constantly shortened, and the number of discovered patterns also increases. This RP-tree-based RP-Growth algorithm is also a prominent algorithm in mining rare itemsets, and FRI-Miner is also based on its efficient performance.

It can be observed that the setting of two thresholds also affects the mining time of the itemset. In addition, normal behavior/patterns often appear in the form of infrequent itemsets, but fewer rare behaviors/patterns can be found with FRI-Miner using fuzzy theory. For example, consider the retail dataset, when setting parameters as *minSup* from 95 to 100, and *maxSup* from 200 to 500, the number of FRIs is changed from 1,261 to 1,593, while the number of RIs can

[1] http://www.philippe-fournier-viger.com/spmf/

Table 4 Number of patterns (candidates and final results) under various parameter settings

		# of patterns under various parameter settings						
		$test_1$	$test_2$	$test_3$	$test_4$	$test_5$	$test_6$	$test_7$
(a) accidents	*minSup*	100	100	100	100	200	200	200
	maxSup	250	300	350	400	250	300	350
	#FRIs	6,638	10,791	17,966	29,032	837	1,516	2,293
	#RIs	387,624	539,064	2,924,992	3,697,552	500	1,656	28,868
(b) foodmart	*minSup*	5	5	5	10	10	10	10
	maxSup	13	15	17	13	15	17	20
	#FRIs	4,514	7,905	9,525	4,263	7,654	9,274	10,031
	#RIs	930	1,223	1,402	554	847	1,026	1,145
(c) chess	*minSup*	100	100	100	150	150	150	150
	maxSup	200	300	400	200	300	400	500
	#FRIs	1,448	18,780	99,014	187	2,013	12,886	59,374
	#RIs	16,414,992	50,739,408	675,273,704	122,040	1,309,176	64,897,692	237,222,940
(d) kosarak	*minSup*	1000	1000	1000	1500	1500	1500	1500
	maxSup	3000	3500	4000	3000	3500	4000	4500
	#FRIs	2,242	2,395	2,546	801	882	977	1,052
	#RIs	642,536	654,922	661,168	195,507	198,575	201,007	202,013
(e) mushroom	*minSup*	100	100	100	150	150	150	150
	maxSup	200	300	400	200	300	400	500
	#FRIs	1,344	5,862	15,405	243	1,185	3,062	5,185
	#RIs	364,992	1,269,244	1,915,464	360,512	475,200	882,000	2,383,752
(f) retail	*minSup*	95	95	95	95	100	100	100
	maxSup	200	300	400	500	200	300	400
	#FRIs	1,261	1,581	1,734	1,795	1,128	1,441	1,593
	#RIs	1,740	3,244	3,887	4,378	1,493	2,929	3,552

reach 1,740 to 3,552. It is clear that the two types of patterns, FRIs and RIs, are different, and the fuzzy-based patterns have more useful meaning. Thus, the concept of FRI has excellent fuzzy modeling capabilities and has a linguistic meaning.

5.2 Efficiency w.r.t. runtime

Runtime vs. support The effects of the *minSup* and *maxSup* thresholds were evaluated first. For each dataset, we uniquely specified the corresponding thresholds. When the minimum support threshold we choose continues to increase and the maximum support threshold remains unchanged, we find that the running time of FRI-Miner will continue to decrease. The results of the detailed experiments are shown in Fig. 7a–f. Similarly, when the minimum support threshold is unchanged and the maximum support threshold continuously increases, the running time continuously increases as the mining range increases. These results are shown in Fig. 6a–f. For RP-Growth previously studied, it is a mining rare itemsets algorithm based on the RP-tree structure. This is especially true when *minSup* is set to be very small, and the runtime increases sharply. For example, in Figs. 6 and 7, the running time of the two compared algorithms always increases with an increase in *maxSup*, and the running time of the two compared algorithms always increases with a decrease in *minSup*. This result corresponds to the following experiments.

Runtime vs. size/density of dataset After the runtime analysis on the benchmark datasets, we found that the size of the tested dataset also had a significant impact on the running time of the experiments. For a larger dataset, we need to spend more time on the mining process. In addition, we found that, for a dense dataset, although the dataset is not large, it takes a longer time in the mining process.

Discussion Based on the results of the datasets in the experimental run, we found that the running time of the FRI-Miner algorithm was longer than that of the RP-Growth algorithm. The reason is that in the process of mining rare itemsets, we need to fuzzify the quantitative values; thus, the running time is relatively longer. Another reason for the longer running time is that FRI-Miner considers that the combination

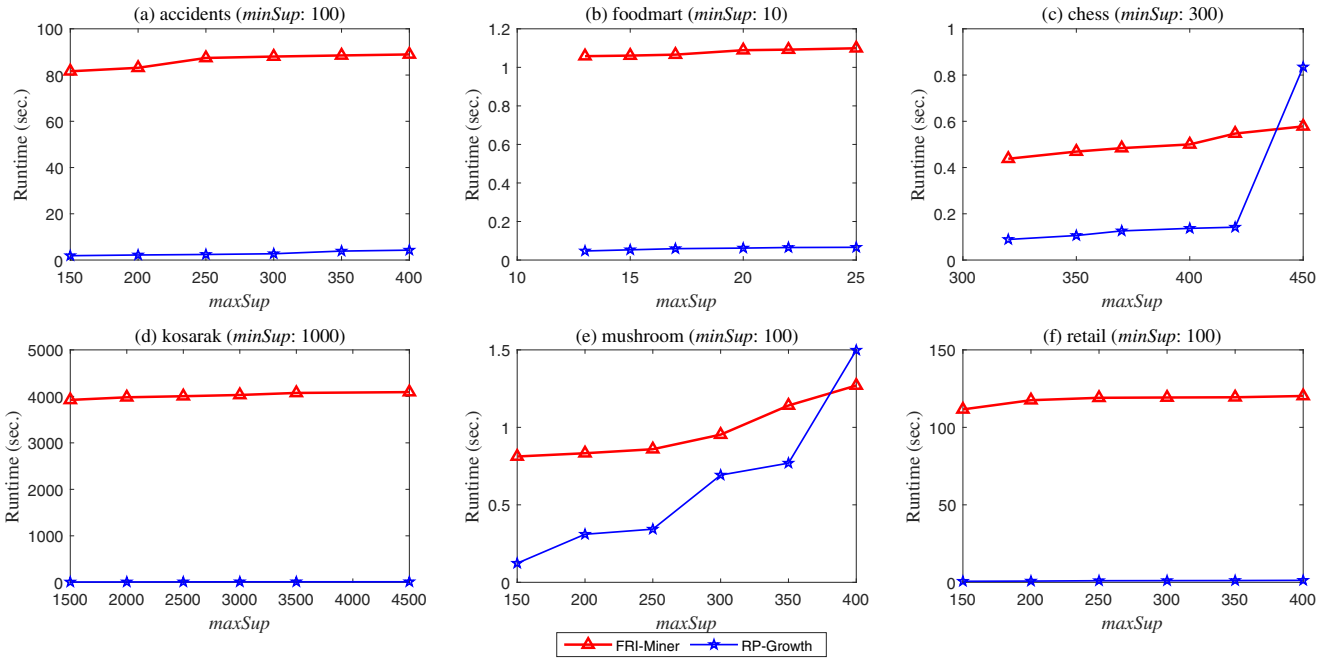


Fig. 6 Runtime vs. *maxSup*

of frequent itemsets may also be rare itemsets when mining rare itemsets. In other words, the branches of frequent itemsets are not removed in the pruning stage, which makes we find more rare itemsets that are ignored. Owing to the existence of pruning strategies, FRI-Miner not only reduces the search space, but also improves the efficiency of the mining task.

5.3 Efficiency w.r.t. pattern

Pattern In this subsection, we compare the number of patterns discovered by the algorithms. Our intuition suggests that in the FRI-Miner algorithm, the number of patterns mined is greater than that of the RP-Growth algorithm. This has also been confirmed in some datasets in the experiment,

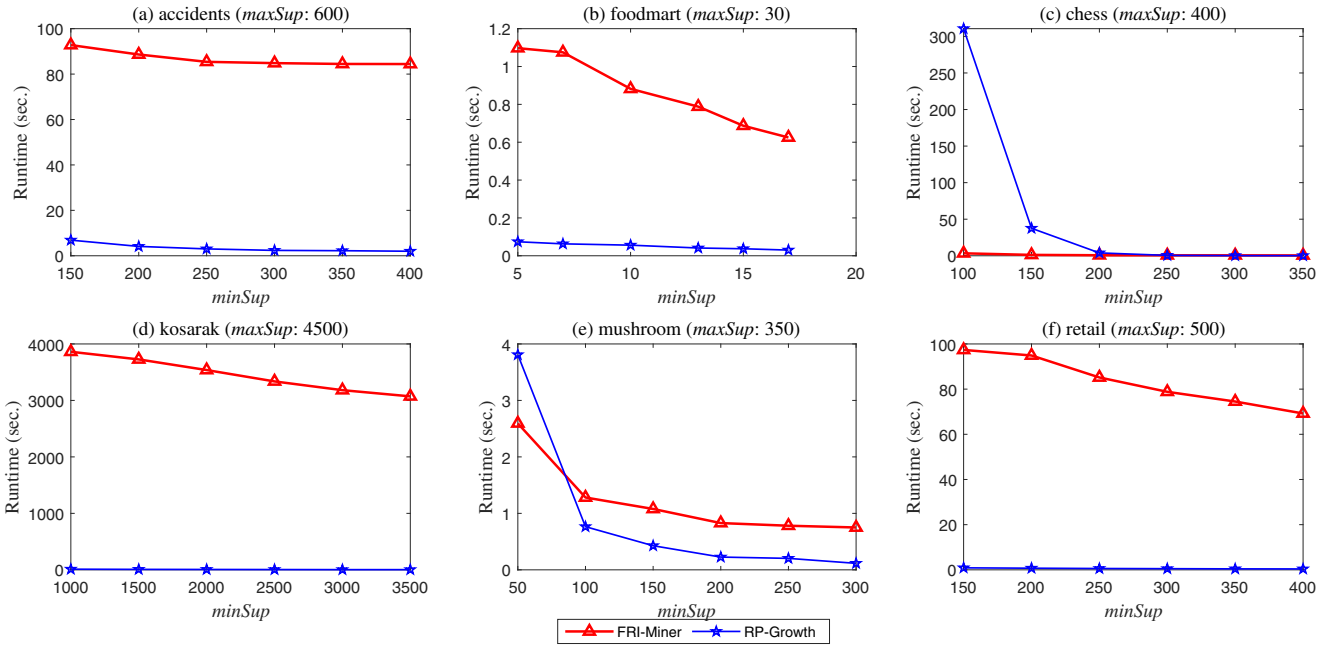


Fig. 7 Runtime vs. *minSup*

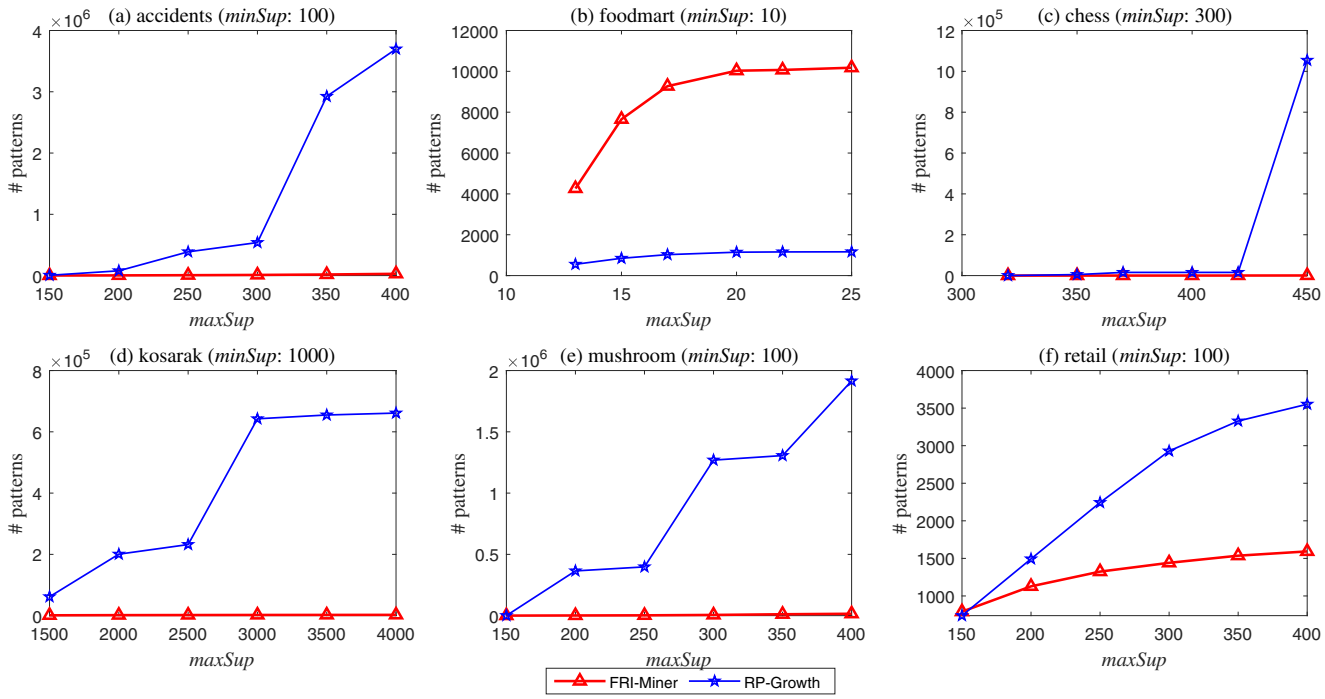


Fig. 8 Pattern vs. *maxSup*

such as dataset accidents and BMS, which well reflect this phenomenon. The results of the experiment may be unclear or conflicting. The reason for this experimental phenomenon is that we use quantitative values and thus use fuzzification and pruning strategies to prune many inconsistent itemsets in the mining process. This pruning strategy ensures that the number of itemsets we finally discover is relatively small, but the practicability of the itemsets that we discover is higher. In the experiments, we found that most datasets reflect this phenomenon, including retail, chess, mushroom,

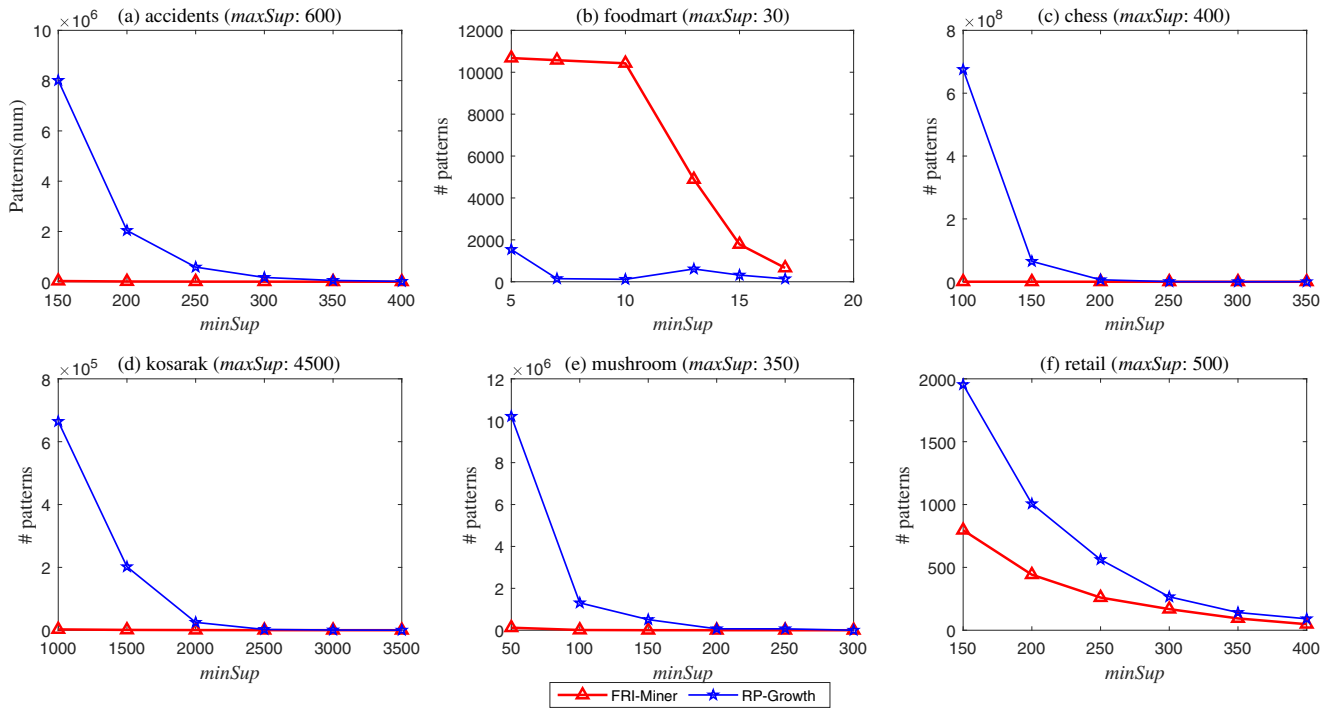


Fig. 9 Pattern vs. *minSup*

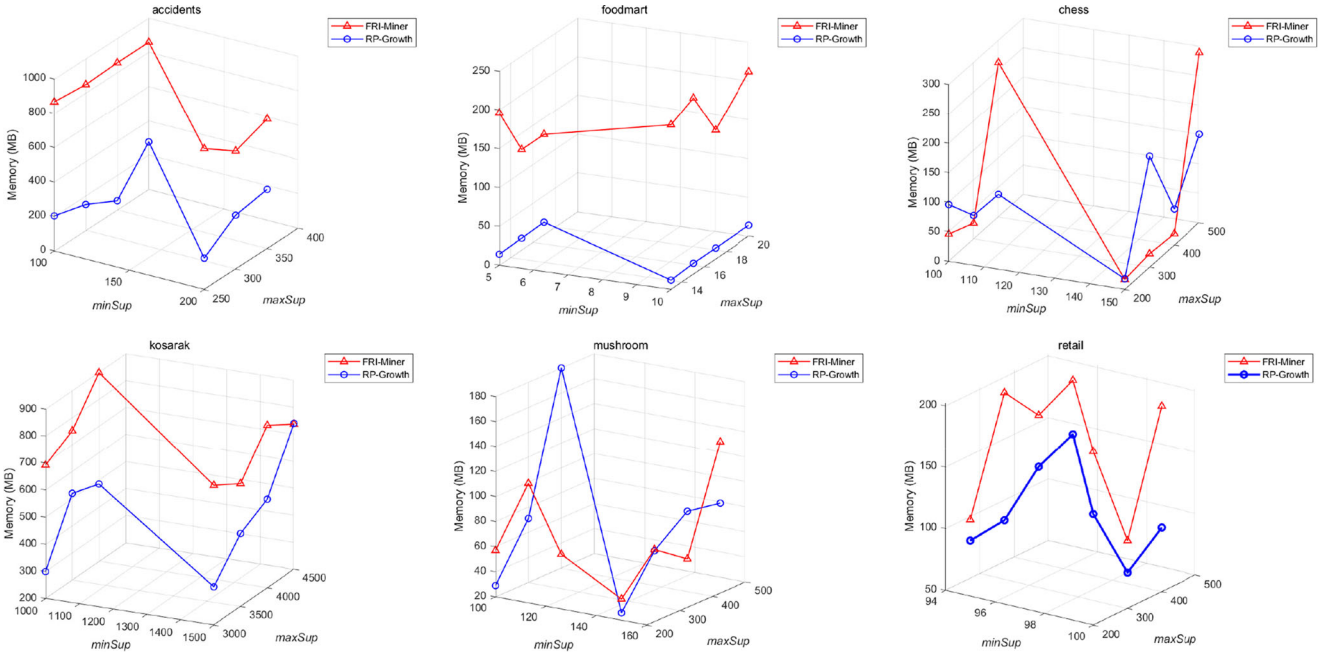


Fig. 10 Memory vs. *minSup*/*maxSup*

and kosarak. The specific experimental results are listed in Table 4.

Pattern vs. support For each dataset, we choose a threshold that meets their own thresholds; when the minimum support threshold selected continues to increase and the maximum support threshold remains unchanged, we find that the number of itemsets extracted will continue to decrease. Figure 9 shows the experimental results. Similarly, when the minimum threshold remains unchanged and the maximum threshold value continuously increases, we find that the number of itemsets mined continuously increases. The experimental results are presented in Fig. 8. This is especially true when *minSup* is set to be very small, and the runtime increases sharply. The effects of the *minSup* and *maxSup* thresholds were evaluated first. For example, in Figs. 8 and 9, the number of patterns of the two compared algorithms always increases with an increase in *maxSup*, and the number of patterns of the two compared algorithms always increases with a decrease in *minSup*.

5.4 Efficiency w.r.t. memory

In this subsection, we further analyze the maximum memory occupied during the mining processes of the algorithms. According to the experimental results of six datasets, we found that in sparse datasets (e.g., retail, kosarak), the FRI-Miner algorithm can extract meaningful rare itemsets, that is, ignore most of the itemsets that do not meet the actual situation, but also the memory usage is lower than that of RP growth. However, for relatively dense datasets, a relatively large memory space is required during the mining process. In the dataset BMS, FRI-Miner shows good performance, not only mining more eligible itemsets but also taking up relatively less memory space. The experimental results of these datasets are shown in Fig. 10a–f.

6 Conclusion and future work

In this paper, we propose a novel fuzzy-theoretic-based algorithm for mining FRIs determined by fuzzy set theory and pattern mining. The purpose of this effective mining algorithm is to find meaningful rare itemsets that meet the minimum thresholds. In contrast to existing algorithms, an effective FRI-Miner algorithm is proposed based on a fuzzy list structure. The algorithm requires specifying two minimum support thresholds; it utilizes several pruning strategies to prune unqualified itemsets and discovers the complete set of rare itemsets containing rare and frequent items. The experimental analysis shows that this algorithm performs well and has an improved overall mining quality compared to that of the existing algorithm.

There are still some limitations in our research, such as the membership function and the specified minimum thresholds, which are defined in advance. If the threshold is specified to be extremely small, many candidate itemsets are generated, which occupy a large storage space. If it is specified as extremely high, some meaningful itemsets are ignored, which results in poor mining quality. It is

hoped that there will be a more convenient algorithm in the future, which can automatically detect the most suitable threshold value and the setting of the membership function to effectively discover rare patterns.

Acknowledgments Thanks for the anonymous reviewers for their insightful comments, which improved the quality of this paper. This work was partially supported by the National Natural Science Foundation of China (Grant Nos. 62002136, 61932011, and 11974373), and Guangzhou Basic and Applied Basic Research Foundation, and Guangdong Basic and Applied Basic Research Foundation (Grant No. 2019B1515120010).

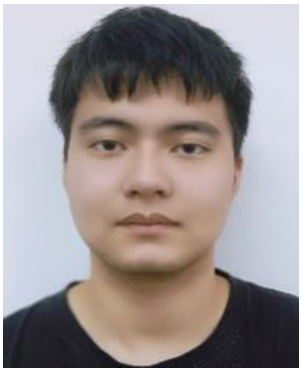
References

1. Agrawal R, Srikant R et al (1994) Fast algorithms for mining association rules. In: 20th International Conference on Very Large Data Bases, pp 487–499
2. Gan W, Lin JC-W, Chao H-C, Zhan J (2017) Data mining in distributed environment: a survey. Wiley Interdiscip Rev Data Min Knowl Discov 7(6):e1216
3. Han J, Pei J, Kamber M (2011) Data mining: concepts and techniques. Elsevier
4. Berry MJA, Linoff GS (2004) Data mining techniques: for marketing, sales, and customer relationship management. Wiley
5. Linoff GS, Berry MJA (2011) Data mining techniques: for marketing, sales, and customer relationship management. Wiley
6. Shaw MJ, Subramaniam C, Tan GW, Welge ME (2001) Knowledge management and data mining for marketing. Decis Support Syst 31(1):127–137
7. Lin JC-W, Gan W, Fournier-Viger P, Hong T-P (2015) RWFIM: Recent weighted-frequent itemsets mining. Eng Appl Artif Intell 45:18–32
8. Fournier-Viger P, Zhang Y, Lin JC-W, Fujita H, Koh YS (2019) Mining local and peak high utility itemsets. Inf Sci 481:344–367
9. Gan W, Lin C-W, Fournier-Viger P, Chao H-C, Tseng V, Yu P (2021) A survey of utility-oriented pattern mining. IEEE Trans Knowl Data Eng 33(4):1306–1327
10. Gan W, Lin JC-W, Fournier-Viger P, Chao H-C, Hong T-P, Fujita H (2018) A survey of incremental high-utility itemset mining. Wiley Interdiscip Rev Data Min Knowl Discov 8(2):e1242
11. Nguyen LTT, Vu VV, Lam MTH, Duong TTM, Manh LT, Nguyen TTT, Vo B, Fujita H (2019) An efficient method for mining high utility closed itemsets. Inf Sci 495:78–99
12. Kim C, Lim J-H, Ng RT, Shim K (2007) SQUIRE: Sequential pattern mining with quantities. J Syst Softw 80(10):1726–1745
13. Le T, Nguyen A, Huynh B, Vo B, Pedrycz W (2018) Mining constrained inter-sequence patterns: a novel approach to cope with item constraints. Appl Intell 48(5):1327–1343
14. Srikant R, Agrawal R (1996) Mining sequential patterns: Generalizations and performance improvements. In: International Conference on Extending Database Technology. Springer, pp 1–17
15. Kiran RU, Reddy PK (2011) Novel techniques to reduce search space in multiple minimum supports-based frequent pattern mining algorithms. In: Proceedings of the 14th International Conference on Extending Database Technology, pp 11–20
16. Koh YS, Ravana SD (2016) Unsupervised rare pattern mining: a survey. ACM Trans Knowl Discov Data 10(4):1–29
17. Szathmary L, Napoli A, Valtchev P (2007) Towards rare itemset mining. In: 19th IEEE International Conference on Tools with Artificial Intelligence. IEEE, pp 305–312
18. Tsang S, Koh YS, Dobbie G (2011) RP-Tree: rare pattern tree mining. In: International Conference on Data Warehousing and Knowledge Discovery. Springer, pp 277–288
19. Zadeh LA (1965) Fuzzy sets. Inf Control 8(3):338–353
20. Chan KCC, Au W-H (1997) Mining fuzzy association rules. In: Proceedings of the 6th International Conference on Information and Knowledge Management, pp 209–215
21. Kuok CM, Fu A, Wong MH (1998) Mining fuzzy association rules in databases. ACM SIGMOD Record 27(1):41–46
22. Hong T-P, Kuo CS, Chi SC (1999) A data mining algorithm for transaction data with quantitative values. Intell Data Anal 3(5):363–376
23. Lin C-W, Hong T-P, Lu W-H (2010) Linguistic data mining with fuzzy FP-trees. Expert Syst Appl 37(6):4560–4567
24. Lin C-W, Hong T-P, Lu W-H (2010) An efficient tree-based fuzzy data mining approach. Int J Fuzzy Syst 12(2):150–157
25. Lin C-W, Hong T-P (2014) Mining fuzzy frequent itemsets based on ubffp trees. J Intell Fuzzy Syst 27(1):535–548
26. Lin JC-W, Li T, Fournier-Viger P, Hong T-P (2015) A fast algorithm for mining fuzzy frequent itemsets. J Intell Fuzzy Syst 29(6):2373–2379
27. Hong T-P, Kuo C-S, Chi S-C (1999) Mining association rules from quantitative data. Intell Data Anal 3(5):363–376
28. Papadimitriou S, Mavroudi S (2005) The fuzzy frequent pattern tree. In: The WSEAS International Conference on Computers, pp 1–7
29. Han J, Pei J, Yin Y, Mao R (2004) Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data Min Knowl Disc 8(1):53–87
30. Gan W, Lin JCW, Fournier-Viger P, Chao HC, Zhan J (2017) Mining of frequent patterns with multiple minimum supports. Eng Appl Artif Intell 60:83–96
31. Ji Y, Ying H, Tran J, Dews P, Mansour A, Massanari RM (2012) A method for mining infrequent causal associations and its application in finding adverse drug reaction signal pairs. IEEE Trans Knowl Data Eng 25(4):721–733
32. Sadhasivam KSC, Angamuthu T (2011) Mining rare itemset with automated support thresholds. J Comput Sci 7(3):394
33. Liu B, Hsu W, Ma Y (1999) 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
34. Lin JC-W, Gan W, Fournier-Viger P, Hong T-P, Zhan J (2016) Efficient mining of high-utility itemsets using multiple minimum utility thresholds. Knowl-Based Syst 113:100–115
35. Gan W, Lin JCW, Fournier-Viger P, Chao HC, Yu PS (2021) Beyond frequency: Utility mining with varied item-specific minimum utility. ACM Trans Internet Technol 21(1):1–32
36. Koh YS, Rountree N (2005) Finding sporadic rules using apriori-inverse. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp 97–106
37. Troiano L, Scibelli G, Birtolo C (2009) A fast algorithm for mining rare itemsets. In: Ninth International Conference on Intelligent Systems Design and Applications. IEEE, pp 1149–1155
38. Hemalatha CS, Vaidehi V, Lakshmi R (2015) Minimal infrequent pattern based approach for mining outliers in data streams. Expert Syst Appl 42(4):1998–2012
39. Huang D, Koh YS, Dobbie G (2012) Rare pattern mining on data streams. In: International Conference on Data Warehousing and Knowledge Discovery. Springer, pp 303–314
40. Huang DTJ, Koh YS, Dobbie G, Pears R (2014) Detecting changes in rare patterns from data streams. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp 437–448

41. Rymon R (1992) Search through systematic set enumeration. In: Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning, pp 539–550
42. Lin JC-W, Gan W, Fournier-Viger P, Hong T-P, Chao H-C (2017) FDHUP: Fast algorithm for mining discriminative high utility patterns. Knowl Inf Syst 51(3):873–909

Yanling Cui received the B.S. degree in Computer Science from Qilu University of Technology (Shandong Academy of Sciences), Shandong, China in 2019. She is currently a postgraduate with the College of Computer Science and Technology, Shandong University of Science and Technology, Shandong, China. Her research interests include data mining, utility computing, and big data.

Wensheng Gan received the B.S. degree in Computer Science from South China Normal University, Guangdong, China in 2013. He received the Ph.D. in Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Guangdong, China in 2019. He was a joint Ph.D. student with the University of Illinois at Chicago, Chicago, IL, USA, from 2017 to 2019. He is currently an Association Professor with the College of Cyber Security, Jinan University, Guangzhou, China. His research interests mainly focus on data mining, big data analytics, and blockchain. He has published more than 80 research papers in peer-reviewed journals (i.e., IEEE TKDE, IEEE TCYB, IEEE TFS, ACM TKDD, ACM TOIT, ACM TDS, ACM TMIS) and international conferences. He is an Associate Editor of Journal of Internet Technology.



Hong Lin is currently an undergraduate in the School of Computer at Guangdong University of Technology. His research interests include data mining and blockchain security.

Weimin Zheng received the Ph.D. degree from the Harbin Institute of Technology, Heilongjiang, China in 2001. He is currently a full Professor with the College of Computer Science and Technology, Shandong University of Science and Technology, Shandong, China. His current research interests include artificial Intelligence, network security, and blockchain.