Research Article

FI-FG: Frequent Item Sets Mining from Datasets with High Number of Transactions by Granular Computing and Fuzzy Set Theory

Zhong-jie Zhang, Jian Huang, and Ying Wei

College of Mechatronics Engineering and Automation, National University of Defense Technology, Changsha 410073, China

Correspondence should be addressed to Jian Huang; 597289799@qq.com

Received 3 July 2015; Revised 7 November 2015; Accepted 23 November 2015

Academic Editor: Anna M. Gil-Lafuente

Copyright © 2015 Zhong-jie Zhang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mining frequent item set (FI) is an important issue in data mining. Considering the limitations of those exact algorithms and sampling methods, a novel FI mining algorithm based on granular computing and fuzzy set theory (FI-GF) is proposed, which mines those datasets with high number of transactions more efficiently. Firstly, the granularity is applied, which compresses the transactions to some granules for reducing the scanning cost. During the granularity, each granule is represented by a fuzzy set, and the transaction scale represented by a granule is optimized. Then, fuzzy set theory is used to compute the supports of item sets based on those granules, which faces the uncertainty brought by the granularity and ensures the accuracy of the final results. Finally, Apriori is applied to get the FIs based on those granules and the new computing way of supports. Through five datasets, FI-GF is compared with the original Apriori to prove its reliability and efficiency and is compared with a representative progressive sampling way, RC-SS, to prove the advantage of the granularity to the sampling method. Results show that FI-GF not only successfully saves the time cost by scanning transactions but also has the high reliability. Meanwhile, the granularity has advantages to those progressive sampling methods.

1. Introduction

Frequent item sets (FIs) contain the items which always appear together in a dataset with the frequency over a specified minimum support [1, 2]. For instance, given a dataset which contains the records of a store, if products A and B are found always bought together by customers, and the frequency of this phenomenon is over the minimum support, $\{A, B\}$ can be seen as a FI. FI can be classified into the quantitative FI (QFI) and the binary FI (BFI). QFI is mined from the quantitative dataset, where every item has a scale of value. BFI is mined from the binary dataset, where every item only has two states, presence or absence. For example, given a dataset which has four items, A, B, C , and D , a QFI may look like $\{A \in (0 \sim 1), B \in (-1 \sim 0.5), C \in (-2 \sim 1.5), D \in (-1 \sim 3.5)\}$, and a BFI may look like $\{A, C\}$ or $\{B, C, D\}$. However, in most cases, when mining FIs, those quantitative datasets are always transformed into binary datasets firstly by the discretization,

so in this thesis, we mainly focus on FI mining from the binary dataset.

FI mining is the stone of association rules mining and many other fields. Many algorithms have been presented in this realm, where Apriori and FP-growth are the most two famous methods of them. Apriori is the most widely applied algorithm [3], which mines all the FIs from a dataset by several loops. In the k th loop, Apriori mines FIs with lengths k , where the FIs got from the $k - 1$ th loop are combined to make the candidate FIs of the k th loop firstly, and Apriori scans the dataset to check every candidate FI and remove the fake candidates. The most serious problem of Apriori is the repeatedly scanning of dataset. If the transaction scale of a dataset is large, the mining speed becomes low. Therefore, another algorithm, FP-growth, is proposed to solve this problem [4]. In FP-growth, the whole dataset is firstly transformed to a complex data structure, and algorithm only scans this data structure but the original dataset, which saves much

time. However, problem also exists. FP-growth consumes too much memory to save this big data structure, and the transformation from dataset to data structure is also very complex.

Furthermore, many other algorithms are also proposed in this field [1, 5]; these algorithms improve the Apriori and FP-growth through removing the minimum support, using the dynamically allocated memory, improving the data structure, and so on, but none of them solve the problem brought by the large transaction scale well.

By researching the above algorithms, we can find that all of them are the exact algorithm, whose aims are to mine the exact results out. However, in some applications of FI mining, the accuracy is much less important than the speed. Therefore, there is a feasible idea which can be used to solve the problem brought by the scanning, and the main thought of it is to sacrifice some accuracy of the results and to earn the faster speed of algorithm [6].

According to this idea, some sampling methods are presented [7–11], in which the original dataset is firstly sampled randomly, and then, the algorithm mines the results just from those samples but the whole dataset, which can extremely cut the cost. However, this solution also brings another problem. Despite the kernel thought of those sampling ways is to enhance the speed through sacrificing some accuracy, the loosing of accuracy also should be controlled in a reasonable range. Generally, there are 3 kinds of ways to achieve this goal, setting the bound of sample scale, estimating the accuracy of results, and progressive sampling.

The bound of sample scale is the shortest or the largest sample scale, which announces the safe range of the sample size. Those bounds are always got by some mathematical inferences and hypothesis, such as a certain distribution of the value of items. Some researches have been done. For example, Zaki and Chakaravarthy both studied the bound of sample size with their teams [7, 8]. However, in consideration of the difference among datasets, the mathematical inferences and hypothesis, which are used to make the bound, do not always apply, so the risk to set a wrong bound is very high.

Unlike setting the bound of sample size, another method estimates the possibility of every FI mined from the sample being the FI mined from the original dataset. For example, Toivonen studied a method, which builds a candidate of FI with these probabilities based on the sample size [9]. Nevertheless, the estimation of the accuracy just offers a reference to the user, which does not give a method to reduce the error probability.

Being different from the above two ways, progressive sampling not only provides a reliable way to control the loosing of accuracy but also avoids a lot of mathematical hypothesis, in which the sample scale keeps changing until the stop condition is satisfied. For example, Parthasarathy presents a progressive way, which keeps increasing or decreasing the sample size until the stopping conditions are satisfied [10]. Chen and his colleagues proposed a method with 2 phases, which progressively builds an appropriate sample [11]. But, the difficulty of these progressive sampling ways is also apparent, which is the difficulty to build an appropriate stop condition. A simple stop condition cannot

ensure the accuracy of the final results well, and the complex stop condition may consume too much time and resources.

Considering the limits of sampling way, it is necessary to come up with a new method, which not only can reduce the cost of scanning dataset but also has an efficient mechanism to ensure the accuracy of results, where the efficiency means that the computing of this mechanism is simple, the quality of this mechanism is high, and it is suitable to different datasets.

Therefore, a novel FI mining algorithm, where the scale of transactions is cut by the granularity and the support is computed by fuzzy set theory, called FI-GF, is proposed in this paper.

The definitions of granular computing are varied [12]. Generally, it is a way to solve problem in different levels through different kinds of information granules. In most cases, granule is a collection of data, which can be built by clustering, partition, and so on and can be denoted by ordinary sets, fuzzy sets, rough sets, and so on [13, 14]. In FI-GF, it is built by the partition of the set of transactions and is represented by fuzzy set.

Fuzzy set describes uncertainty, where every element has a membership to represent the degree to which it is contained [15]. Fuzzy set theory is applied widely to the FI mining from those datasets with uncertainty [16–19]. Considering that the granule is denoted by fuzzy set and the granularity may bring uncertainty, fuzzy set theory is joined in FI-GF.

In short, the new proposed algorithm, FI-GF, has the following innovations and advantages:

- (1) The granularity is used to cut the transaction scale by compressing transactions to granules, and compared with progressive sampling, it is more efficient.
- (2) Fuzzy set is used to denote granules, and a method to calculate the supports of item sets based on those granules is designed, which helps the algorithm to deal with the uncertainty brought by transaction reduction well.

2. Basic Concepts

2.1. Frequent Item Sets Mining. A formal specification of the problem is presented as follows: Let $\mathbf{I} = \{i_1, i_2, \dots, i_m\}$ be a set of m distinct items. A dataset $\mathbf{D} = \{\mathbf{t}_1, \mathbf{t}_2, \dots, \mathbf{t}_n\}$ is a set of n transactions, in which $\mathbf{t}_i \subseteq \mathbf{I}$. A set $\mathbf{X} \subseteq \mathbf{I}$ is called an item set. $|\mathbf{X}|$, called the length of $\mathbf{X}$, is the number of items in $\mathbf{X}$. Item sets whose length is l are called l -item-sets. The support of $\mathbf{X}$, denoted by $\text{sup}(\mathbf{X})$, is the fraction of $\mathbf{D}$ containing $\mathbf{X}$.

$\mathbf{X}$ is a FI if and only if $\text{sup}(\mathbf{X})$ is larger than a specific minimum support, denoted by min_sup , which indicates that the presence of $\mathbf{X}$ is significant. Given min_sup , the goal of FI mining is to enumerate all the item sets whose supports are higher than min_sup [20–22].

2.2. Granular Computing. Given a problem or a system $\mathbf{Sys} = \{e_1, e_2, \dots, e_n\}$, where e is the basic element of $\mathbf{Sys}$, the granular computing can be described by

$$\begin{aligned} \{\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_m\} &= \text{Granularity}(\mathbf{Sys}), \\ \mathbf{R} &= \text{Process}(\{\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_m\}), \end{aligned} \quad (1)$$

where the granularity can be the partition, clustering, and other process of **Sys**, whose kernel is to abstract or define the original problem or system at a different level and study the problem based on this new level. **G** is the granule, which is the basic elements of the problem or system at the new level. In every computation, **G** is the smallest unit, which means that the details inside are ignored after the granularity. In most cases, a granule represents a collection of e , and a granule can be defined as ordinary sets, fuzzy sets, rough sets, and so on. The process based on **G** is variant, which is according to the different definitions of **G**. **R** is the set of results or the set of answers of **Sys** [23–25].

2.3. Fuzzy Set. Fuzzy set is a widely applied tool and theory, which is proposed to represent the uncertainty concept. A fuzzy set **F** of a reference set **U** is identified by a membership function, which is denoted by $F(u)$ [15]. For $\forall u \in U$, $F(u)$ shows the degree to which u is contained by **F**, and **U** is the universal set. Generally, $F(u) \in [0, 1]$. An ordinary set **S** can be regarded as a fuzzy set with the membership function $S(u) \in \{0, 1\}$. The length of a fuzzy set **F** is

$$|F| = \sum_{u \in U} F(u). \quad (2)$$

To operate fuzzy sets in a formal way, fuzzy set theory offers some operators, where two of the most popular formulas are shown as follows [15, 16]:

$$\begin{aligned} (F \cap G)(u) &= F(u) \wedge G(u), \\ (F \cup G)(u) &= F(u) \vee G(u), \end{aligned} \quad (3)$$

where $a \wedge b = \min(a, b)$ and $a \vee b = \max(a, b)$.

3. The Proposed Algorithm FI-GF

Given a dataset $D = \{t_1, t_2, \dots, t_n\}$, Figure 1 is the simple working diagram of FI-GF. The whole process can be divided into two main parts, which are the granularity part and the mining part. In the granularity part, the dataset is scanned and partitioned from the beginning to end, and meanwhile, those transactions which are neighboring and similar are collected and represented by several granules. Every granule in FI-GF is represented by a fuzzy set, and the reference set, which is denoted by **I**, of these fuzzy sets is the universe set of items of the dataset **D**. In the mining part, the Apriori algorithm is used to mine FIs from those granules, and the support of every item set is computed through fuzzy set theory.

3.1. The Collection of Transactions. Like what Figure 2 shows, the core idea of the granularity is to compress the transaction scale. During the granularity, the dataset is scanned, and those transactions which are similar and neighboring are collected and represented by a fuzzy set.

Therefore, the first problem which should be solved is how to partition the dataset and collect transactions.

Given a collection of transactions **g**, which is used to build a granule G_g , the evaluation of **g** can be based on two factors of G_g [14].

(1) The coverage of granule G_g , denoted by $\text{cov}(G_g)$, is the capability of G_g to contain information. Generally, it is an increasing function of the data scale represented by G_g . In FI-GF, it is defined as

$$\text{cov}(G_g) = -e^{-|g|} + 1. \quad (4)$$

This factor can be easily understood. The purpose of the granularity is to save the time for scanning transactions, so the more transactions a granule can represent, the more time and resources will be saved.

(2) The specificity of granule G_g , denoted by $\text{sp}(G_g)$, is the capability of G_g to precisely represent information. The higher $\text{sp}(G_g)$ is, the more precisely G_g can represent the information in **g**. In this thesis, it is defined as

$$\text{sp}(G_g) = e^{-\text{dif}(g)}, \quad (5)$$

where $\text{dif}(g)$ is the difference among the transactions in **g**, which is calculated as

$$\text{dif}(g) = \sum_{t_j \in g} \text{HD}(t_j, t_{j-1}), \quad (6)$$

where $\text{HD}(t_j, t_{j-1})$ is the hamming distance between t_j and t_{j-1} . Obviously, $\text{sp}(G_g)$ is a decreasing function of the data scale in **g**.

The second factor is also important. Since the granularity is used to compress the transactions, it will certainly destroy and ignore some original information of transactions. The more transactions a granule has, the more differences among transactions are ignored. This factor is to evaluate the degree to which the granule can preserve the original information.

However, in most cases, these two factors are restricted by each other, and an appropriate granule is supposed to not only cover more data but also represent information more precisely. Therefore, the collection of transactions can be formulated as the following optimization problem:

$$\text{Max} \quad \text{cov}(G_g) \cdot \text{sp}(G_g). \quad (7)$$

Furthermore, the weights of coverage and specificity are not always the same, so two parameters are involved to change the optimization problem into (8), where α and β are the parameters to control the importance of coverage and specificity. If α is higher, the granule tends to contain more transactions and relatively disregards of the capability to precisely represent information, and vice versa. Consider

$$\begin{aligned} \text{Max} \quad \text{Col}(G_g) \\ \text{Col}(G_g) = (-e^{-|g|^\alpha} + 1) \cdot e^{-\beta \cdot \sum_{t_j \in g} \text{HD}(t_j, t_{j-1})}. \end{aligned} \quad (8)$$

On the other hand, α and β have another purpose, which is to control the average transaction scale in the granule. Despite the transaction scales in different granules are determined by the balances of coverage and specificity, the average of those scales can be controlled to an expected value. Suppose the average of hamming distance between two transactions in a dataset is AHD; (8) can be approximately regarded as

$$\text{Col}(G_g) = (-e^{-|g|^\alpha} + 1) \cdot e^{-\beta \cdot |g| \cdot \text{AHD}}. \quad (9)$$

FIGURE 1: The simple working diagram of FI-GF.

FIGURE 2: The granularity of transactions.

Then, the expected $|g|$, which can make Col get to the peak, can also be estimated, which is

$$|g| = -\frac{1}{\alpha} \cdot \ln \frac{\beta \cdot \text{AHD}}{\alpha + \beta \cdot \text{AHD}}, \quad (10)$$

and AHD can be estimated from the sample of transactions. Therefore, through controlling the value of α and β , the average transaction scale contained by a granule can be also controlled. Generally, if both α and β become smaller, the average transaction scale in a granule becomes larger, and if both α and β become larger, the average scale becomes smaller, which can be deduced from the above approximate function.

Generally, when $|g|$ is growing, the curves of $\text{cov}(G_g)$, $\text{sp}(G_g)$, and $\text{Col}(G_g)$ look like Figure 3. Firstly, the information of the granule is little, and the differences of transactions in the granule are relatively few, so sp is high and cov is low. Then, with the growing of transaction scale, the granule contains more information while bringing more differences at the same time, so cov increases but sp decreases. For Col, at the beginning, the capability to contain more information

FIGURE 3: The example of the curves of cov, sp, and Col.

and the capability to precisely represent the information of the granule are not balanced, so Col is low initially. Then, with the increase of transaction scale, the balance of cov and sp becomes better, so Col increases too. However, after the peak of Col, the information in the granule is enough but the differences among different transactions make the granule hard to precisely represent the information, so the balance of cov and sp becomes worse, and Col starts to decrease. This phenomenon is verified and shown in Experiment 1 through 5 datasets.

To sum up, the stop condition of a collection is whether the current value of $\text{Col}(\mathbf{G}_g)$ is near the optimal value or not. Considering that, after $\text{Col}(\mathbf{G}_g)$ passes the peak, its tendency becomes decreasing, so the stop condition of a collection is designed as whether the following inequality is satisfied. Consider

$$\text{Stop}(\mathbf{G}_g) \leq \sigma, \quad (11)$$

where

$$\text{Stop}(\mathbf{G}_g) = \text{Col}(\mathbf{G}_g) - \text{Col}(\mathbf{G}_{g - \{t_{|g|}, t_{|g|-1}, \dots, t_{|g|-q}\}}). \quad (12)$$

$\mathbf{g} - \{t_{|g|}, t_{|g|-1}, \dots, t_{|g|-q}\}$ is the collection of transactions which removes $\{t_{|g|}, t_{|g|-1}, \dots, t_{|g|-q}\}$ from $\mathbf{g}$, and t_i is the i th transaction which is put into $\mathbf{g}$. Therefore, $\text{Stop}(\mathbf{G}_g)$ is used to evaluate the growth rate of Col. Parameter σ is the value to help decide if Col is near the peak. $\text{Stop}(\mathbf{G}_g) \leq \sigma$ means that the growth rate of Col is low enough, and Col is probably near the peak, so this collection can be stopped. Parameter q is used to keep the calculation of the growth rate of Col from the fluctuation, which is brought by the fluctuation of $\text{HD}(t_j, t_{j-1})$. The growth rate of Col is not a smooth curve.

FIGURE 4: The process to make a collection into a granule.

If q is high, the effect of fluctuation will be reduced into a low level, but the algorithm will respond more slowly to the coming of peak, and vice versa.

3.2. Representing the Collection by a Granule. After the partition of dataset, we need to design a way to represent those collections which contains transactions and as is shown in Figure 4; fuzzy set is used by FI-GF to do this job, whose form based on Zadeh's [15] way is shown as

$$\mathbf{G}_g = \sum_{i \in \mathbf{I}} \frac{\mathbf{G}_g(i)}{i}, \quad (13)$$

in which $\mathbf{I}$ is the universe set of items of the dataset $\mathbf{D}$ and $\mathbf{G}_g(i)$ is the membership degree of the item i in $\mathbf{G}_g$. Actually, $\mathbf{G}_g$ can be seen as a simplification of the collection $\mathbf{g}$, and it is used to reduce the cost of scanning dataset.

The problem that emerges here is how to determine $\mathbf{G}_g(i)$ of every item in $\mathbf{I}$. The fuzzy statistic test is used to do this job in FI-GF [26].

If the granule $\mathbf{G}_g$ is regarded as a concept, a transaction $t \in \mathbf{g}$ can be seen as a sample of $\mathbf{G}_g$. Then, the membership degree of an item i in $\mathbf{G}_g$ can be determined by

$$\mathbf{G}_g(i) = \frac{\sum_{t \in \mathbf{g}} \text{incl}(\{i\}, t)}{|\mathbf{g}|}, \quad (14)$$

where $\text{incl}(\mathbf{a}, \mathbf{b}) = 1$ if $\mathbf{a} \subseteq \mathbf{b}$ or $\text{incl}(\mathbf{a}, \mathbf{b}) = 0$.

For example, given a granule $\mathbf{G}_{g'}$, where $\mathbf{g}'$ is the collection of transactions which will be represented by $\mathbf{G}_{g'}$, if $\mathbf{g}' = \{t_1 = \{i_1, i_3, i_4\}, t_2 = \{i_1, i_2\}, t_3 = \{i_2, i_4\}\}$, we can know that

$$\begin{aligned} \text{incl}(\{i_1\}, t_1) &= 1, \\ \text{incl}(\{i_1\}, t_2) &= 1, \\ \text{incl}(\{i_1\}, t_3) &= 0, \end{aligned} \quad (15)$$

so $\mathbf{G}_{g'}(i_1) = (1 + 1 + 0)/3 = 2/3$, and so on. Therefore,

$$\begin{aligned} \mathbf{G}_{g'} &= \frac{\mathbf{G}_{g'}(i_1)}{i_1} + \frac{\mathbf{G}_{g'}(i_2)}{i_2} + \frac{\mathbf{G}_{g'}(i_3)}{i_3} + \frac{\mathbf{G}_{g'}(i_4)}{i_4} \\ &\approx \frac{0.67}{i_1} + \frac{0.67}{i_2} + \frac{0.33}{i_3} + \frac{0.67}{i_4}. \end{aligned} \quad (16)$$

To sum up, the pseudo of the granularity is designed and shown as Algorithm 1.


```

Input     $\mathbf{D} = \{t_1, t_2, \dots, t_n\}, \alpha, \beta, \sigma$  and  $q$ .
Output    $\mathbf{GD}$ , the set of granules, and  $\mathbf{ND}$ , the set of the scale of data represented by every granule.
 $\mathbf{GD} = \emptyset, \mathbf{ND} = \emptyset, \mathbf{g} = \emptyset$ .
For  $j = 1 \rightarrow |\mathbf{D}|$ 
   $\mathbf{g} = \mathbf{g} \cup \{t_j\}$ .
  If  $\text{Stop}(\mathbf{G}_g) \leq \sigma$ 
    Using (14) to transform  $\mathbf{g}$  to  $\mathbf{G}_g$ .
     $\mathbf{GD} = \mathbf{GD} \cup \{\mathbf{G}_g\}, \mathbf{ND} = \mathbf{ND} \cup \{|\mathbf{g}|\}, \mathbf{g} = \emptyset$ .
  End If
End For

```

ALGORITHM 1: The granularity.

3.3. The Calculation of Supports Based on Granules. After the granularity, transactions are compressed to a set of granules $\mathbf{GD}$, and much time, which is the cost to scan the transactions, is saved. Next, how to calculate the supports of item sets through those granules should be discussed.

Considering that a granule in FI-GF is a fuzzy set, the calculation of supports in theory of fuzzy association rules can be used for [19], in which, given a set of fuzzy sets $\mathbf{GD}$ and an item set $\mathbf{X} \subseteq \mathbf{I}$, the supports of $\mathbf{X}$ are defined as

$$\text{Fsup}(\mathbf{X}) = \frac{\sum_{\mathbf{G}_g \in \mathbf{GD}} \mathbf{G}_g(i_1) \wedge \dots \wedge \mathbf{G}_g(i_{|\mathbf{X}|})}{|\mathbf{GD}|}, \quad (17)$$

where $i \in \mathbf{X}$.

However, if (17) is applied in FI-GF, two defects will emerge. Firstly, $\text{Fsup}(\mathbf{X})$ only depends on the item of $\mathbf{X}$ with the minimum degree of membership, and the effects of others are ignored. Secondly, in $\text{Fsup}(\mathbf{X})$, the contributions of all the elements in $\mathbf{GD}$ are the same, but the transaction scales represented by different granules are variant. Therefore, another method should be proposed to calculate supports through $\mathbf{GD}$.

Before the granularity, the contribution of a transaction $\mathbf{t}$ to $\text{sup}(\mathbf{X})$ depends on whether $\mathbf{t}$ covers $\mathbf{X}$, so the contribution of a granule to $\text{sup}(\mathbf{X})$ should be the degree to which the granule covers $\mathbf{X}$. According to this thought, we firstly define the degree to which a granule $\mathbf{G}$ covers a fuzzy set $\mathbf{F}$, which is shown as follows:

$$\text{Fincl}(\mathbf{F}, \mathbf{G}) = \frac{\sum_{u \in \mathbf{U}} \mathbf{F}(u) \wedge \mathbf{G}(u)}{\sum_{u \in \mathbf{U}} \mathbf{F}(u)}, \quad (18)$$

where $\mathbf{U}$ is the reference set. Then, the contribution of a granule $\mathbf{G}_g$ to $\text{sup}(\mathbf{X})$ can be defined as $\text{Fincl}(\mathbf{X}^\gamma, \mathbf{G}_g)$, where $\mathbf{X}^\gamma$ is a fuzzy set whose reference set is $\mathbf{I}$. If $i \in \mathbf{X}$, $\mathbf{X}^\gamma(i) = \gamma$ or $\mathbf{X}^\gamma(i) = 0$. The parameter γ determines how strictly the degree to which $\mathbf{G}_g$ covers $\mathbf{X}$ is evaluated. If γ is larger, the evaluation is more strict, and vice versa.

For example, there is a granule which is

$$\mathbf{G}_{g'} = \frac{0.67}{i_1} + \frac{0.67}{i_2} + \frac{0.33}{i_3} + \frac{0.67}{i_4}. \quad (19)$$

FIGURE 5: The contribution of $\mathbf{G}_g$ to $\text{sup}(\mathbf{X})$.

For an item set $\mathbf{X}' = \{i_1, i_3, i_4\}$,
If $\gamma = 0.8$, we can get that

$$\mathbf{X}'^\gamma = \frac{0.8}{i_1} + \frac{0}{i_2} + \frac{0.8}{i_3} + \frac{0.8}{i_4}, \quad (20)$$

$$\text{Fincl}(\mathbf{X}'^\gamma, \mathbf{G}_{g'}) = \frac{0.67 + 0 + 0.33 + 0.67}{0.8 + 0 + 0.8 + 0.8} \approx 0.70.$$

However, if $\gamma = 0.4$, we can get that

$$\text{Fincl}(\mathbf{X}'^\gamma, \mathbf{G}_{g'}) = \frac{0.4 + 0 + 0.33 + 0.4}{0.4 + 0 + 0.4 + 0.4} \approx 0.94. \quad (21)$$

Obviously, $\text{Fincl}(\mathbf{X}'^\gamma, \mathbf{G}_{g'})$ raises when γ becomes smaller. The meaning of $\text{Fincl}(\mathbf{X}'^\gamma, \mathbf{G}_g)$ is shown as the blue area in Figure 5.

Finally, the supports of an item set $\mathbf{X}$ based on $\mathbf{GD}$ are defined as

$$\text{Gsup}(\mathbf{X}) = \frac{\sum_{\mathbf{G}_g \in \mathbf{GD}} (n_{\mathbf{G}_g} \cdot \text{Fincl}(\mathbf{X}^\gamma, \mathbf{G}_g))}{|\mathbf{D}|}, \quad (22)$$

where $n_{\mathbf{G}_g} \in \mathbf{ND}$ is the transaction scales which are represented by $\mathbf{G}_g$.

For example, a dataset is given which only has 2 granules, $\mathbf{G}_{g_1}$ and $\mathbf{G}_{g_2}$, where $|\mathbf{g}_1| = 3$ and $|\mathbf{g}_2| = 2$, and

$$\begin{aligned} \mathbf{G}_{g_1} &= \frac{0.67}{i_1} + \frac{0.67}{i_2} + \frac{0.33}{i_3} + \frac{0.67}{i_4}, \\ \mathbf{G}_{g_2} &= \frac{1}{i_1} + \frac{0.5}{i_2} + \frac{0.5}{i_3} + \frac{0.5}{i_4}. \end{aligned} \quad (23)$$

TABLE 1: Datasets used in experiments.

Dataset	Number of Items	Number of trans.
Kosarak	41270	990002
T10I4D100K	1000	100000
Retail	16470	88162
Connect	130	67557
Mushroom	120	8124

For an item set $\mathbf{X}' = \{i_1, i_3, i_4\}$, we can get that when $\gamma = 0.8$, $\text{Fincl}(\mathbf{X}'^\gamma, \mathbf{G}_{g_1}) \approx 0.70$ and $\text{Fincl}(\mathbf{X}'^\gamma, \mathbf{G}_{g_2}) \approx 0.75$. So, the support of $\mathbf{X}'$ which is calculated based on $\mathbf{G}_{g_1}$ and $\mathbf{G}_{g_2}$ is

$$\begin{aligned} \text{Gsup}(\mathbf{X}') &= \frac{|\mathbf{g}_1| \cdot \text{Fincl}(\mathbf{X}'^\gamma, \mathbf{G}_{g_1}) + |\mathbf{g}_2| \cdot \text{Fincl}(\mathbf{X}'^\gamma, \mathbf{G}_{g_2})}{|\mathbf{g}_1| + |\mathbf{g}_2|} \quad (24) \\ &= \frac{3 \cdot 0.7 + 2 \cdot 0.75}{5} = 0.72. \end{aligned}$$

3.4. The Dynamical Minimum Support. After the granularity, Apriori is used to mine FIs based on those granules and (22). To reduce the candidate results, a dynamical minimum support, denoted by $\text{min_sup}^{(k/l)^\mu}$, is designed to replace the original one, where l is the length of the item sets which will be mined out, k is the current loop number of Apriori, and μ is used to control the change rate of $\text{min_sup}^{(k/l)^\mu}$.

Firstly, if the target FIs are shorter, $\text{min_sup}^{(k/l)^\mu}$ is relatively large, because, compared with the longer one, the shorter item set are always more frequent. Then, if the loop number of Apriori is higher, $\text{min_sup}^{(k/l)^\mu}$ also becomes larger, because the higher loop number causes the longer candidate results and the more complex computation, which means that more useless candidate results should be removed. Through these ways, the dynamical minimum support can efficiently reduce the cost of Apriori.

3.5. The Whole Pseudo of FI-GF. Summary, the pseudo of FI-GF, based on the granularity, $\text{Gsup}(\mathbf{X})$, Apriori, and dynamical minimum support is shown as Algorithm 2.

4. The Experiments and Discussions

Several datasets are used to evaluate FI-GF. They are kosarak, T10I4D100K, retail, connect, and mushroom, which are downloaded from <http://fimi.ua.ac.be/data/> and shown in Table 1. All experiments are run on Matlab under the Windows XP, and the computer which is used by us has a 2.4 GHz CPU and 2.92 GB RAM. All algorithms are coded by m language.

Experiment 1. In Experiment 1, the method of the granularity in FI-GF, which is Algorithm 1, is applied to the datasets in Table 1. The purpose of this experiment is to display the working principle of the granularity and to help us understand what happens when the granules are being generated.

```

Input     $\mathbf{D}, \alpha, \beta, \sigma, q, \gamma, \mu, l$  and  $\text{min\_sup}$ .
Output   The set of frequent  $l$ -item-sets FI.
 $[\mathbf{GD}, \mathbf{ND}] = \text{granularity}(\mathbf{D}, \alpha, \beta, \sigma, q)$  // Algorithm 1
 $\mathbf{L}_1 = \{i \in \mathbf{I} \mid \text{Gsup}(\{i\}) \geq \text{min\_sup}^{(1/l)^\mu}\}$ 
For  $k = 2 \rightarrow l$ 
     $\mathbf{L}_k = \emptyset$ 
     $\mathbf{C}_k = \text{apriori\_gen}(\mathbf{L}_{k-1})$ 
    For each  $\mathbf{X} \in \mathbf{C}_k$ 
        If  $\text{Gsup}(\mathbf{X}) \geq \text{min\_sup}^{(k/l)^\mu}$ 
             $\mathbf{L}_k = \mathbf{L}_k \cup \mathbf{X}$ 
        End if
    End for
End for
FI =  $\mathbf{L}_l$ 
Procedure  $\text{apriori\_gen}(\mathbf{L}_{k-1})$ 
For each  $\mathbf{I}_1, \mathbf{I}_2 \in \mathbf{L}_{k-1}$ 
    If  $\text{HD}(\mathbf{I}_1, \mathbf{I}_2) = 1$ 
         $\mathbf{c} = \mathbf{I}_1 \cup \mathbf{I}_2$ 
        If each  $k-1$  subset  $\mathbf{s}$  of  $\mathbf{c}$  satisfies that  $\mathbf{s} \in \mathbf{L}_{k-1}$ 
             $\mathbf{C}_k = \mathbf{C}_k \cup \mathbf{c}$ 
        End if
    End if
End for
Return  $\mathbf{C}_k$ 

```

ALGORITHM 2: FI-GF.

TABLE 2: Parameters of FI-GF in Experiment 1.

Dataset	α	β	σ	q
Kosarak	0.00034	0.000024	0	500
T10I4D100K	0.0035	0.00018	0	10
Retail	0.0069	0.00049	0	5
Connect	0.0069	0.0012	0	5
Mushroom	0.0347	0.0022	0	5

Table 2 shows the parameter values of Algorithm 1 in Experiment 1, and Figures 6 and 7 exhibit the details of the granularity.

Figure 6 describes how the curves of cov, sp, and Col change when the transaction scales represented by those first granules in the datasets of Table 1 are growing. Cov, sp, and Col, respectively, describe the capability of a granule to contain more information, the capability of a granule to precisely represent information, and the balance of cov and sp. For every subfigure in Figure 6, the horizontal axis is the transaction scale represented by the first granule, and the vertical axis is the value of cov, sp, and Col.

Figure 7 describes the transaction scale of all the granules in the datasets of Table 1 after the granularity. For every subfigure, the horizontal axis represents the sequence numbers of granules, and the vertical axis represents the transaction scale of every granule.

By analyzing Figures 6 and 7, several phenomenon can be got and explained.

(1) According to Figure 6, we can know that when the transaction scale represented by a granule increases,

FIGURE 6: The curves of Col, cov, and sp when FI-GF is building the first granules of the datasets in Table 1.

cov increases and sp decreases. This phenomenon can be qualitatively explained as follows. To begin with, the more transactions a granule represents, the more information it contains, so cov increases along with the transaction scale. Then, because of the differences among transactions, the more transactions a granule represents, the more difficult it is for the granule to precisely represent the information, so sp decreases when the transaction scale goes up. On the other hand, the quantitative explanation can be got from the definition of cov and sp, which is (4) and (5). The function

$\text{conv}(\mathbf{G}_g) = -e^{-|g|} + 1$ increases along with $|g|$ and function $\text{sp}(\mathbf{G}_g) = e^{-\text{dif}(g)}$ decreases along with $|g|$, and $|g|$ is the transaction scale represented by the granule $\mathbf{G}_g$.

(2) Furthermore, according to Figure 6, we can see that every curve of Col goes up at the beginning and decreases after the peak. At the beginning, when the transaction scale is very low, although the granule can precisely represent those transactions, the information contained in it is too little. Therefore, when the transaction scale increases, Col goes up. Then, after the transaction scale grows to a certain level,

FIGURE 7: The numbers of transactions of granules after the granularities of those datasets in Table 1.

the amount of information is enough. At the same time, the differences among transactions become significant, so Col begins to go down.

(3) According to Figures 6 and 7, we can know that the curves of cov , sp , and Col are different in different datasets, and the distributions of the transaction scales which are represented by granules are also different in different datasets. This phenomenon is caused by the diversity of difference in transactions from different datasets. The larger difference in transactions makes sp fall faster and the Col get to the peak earlier, and vice versa. On the other hand, α and β , which, respectively, decide the weight of cov and sp , also affect the shapes of curves. When α increases, the capability to contain more information becomes more important and cov grows

faster, and vice versa. Meanwhile, if β increases, sp drops faster, and Col will get to the peak earlier.

(4) According to Figure 7, it is obvious that, for every dataset, the transaction scales represented by granules are variant. This phenomenon is brought by the variance of differences among transactions in different parts of a dataset. The large difference results in the faster falling of sp and the earlier peak of Col , which limits the scale of transactions, and vice versa.

To sum up, according to Figures 6 and 7, it can be known that, for a granule, the capability to contain more information and the capability to precisely represent the information are restricted by each other. Meanwhile, when the granularity is applied to different datasets or different parts in the same

FIGURE 8: The time cost by Apriori and FI-GF.

dataset, the changes of cov, sp, and Col and the transaction scales of granules are all different. This phenomenon is not only caused by the diversity of difference in transactions from different datasets and different parts of a same dataset but also brought by the setting of α and β in Table 2, which decides the weights of cov and sp.

Experiment 2. In Experiment 2, FI-GF is used to mine FIs from those datasets in Table 1, where FI-GF is compared with the original Apriori to test its reliability and efficiency. To be fair, the dynamical minimum support, $\min_sup^{(k/l)^\mu}$, is also applied to Apriori. The parameters of the granularity in Experiment 2 are the same as Table 2. In addition to this, $l = 3$, and the other parameters are shown in Table 3.

In Figure 8, the time cost by Apriori and FI-GF under $\gamma = 1$ are recorded in a histogram. Parameter γ represents the stringency to evaluate the degree to which a granule contains an item set.

Table 3 describes the results which are mined by Apriori and FI-GF, where $\min_sup$ and μ , respectively, represent the base number and the parameter controlling the change rate of $\min_sup^{(k/l)^\mu}$. In Table 3, 4 groups of results are recorded, which are the results generated by FI-GF when γ are, respectively, set to 0.9, 0.95, and 1 and the results generated by the original Apriori.

From Figure 8 and Table 3, the following can be got.

(1) According to Figure 8, we can know that, compared with Apriori, FI-GF saves a lot of time. For almost every dataset, FI-GF is at least twice more efficient than Apriori. This advantage is mainly caused by the granularity in FI-GF. To mine FIs from a dataset, Apriori has to repeatedly scan the dataset. If the transaction scale of a dataset is very large, too

much time will be costly. However, FI-GF firstly puts all the transactions into some granules, and the algorithm just needs to scan the granules but the original transactions, so FI-GF is more efficient.

(2) According to Table 3, it can be known that the FIs, which are mined out by the original Apriori, can always be mined out by FI-GF too. This phenomenon proves that the uncertainty, brought by the granularity and the calculation of Gsup, has been faced by FI-GF successfully. The novel algorithm can ensure the reliability of the final results.

(3) Sometimes, the number of results, which are mined out by FI-GF, are more than the number of results which are mined out by Apriori. This phenomenon can be explained by the follows. Firstly, the contribution of a granule G_g to the support of an item set X is defined by $\text{Fincl}(X', G_g)$. This formula describes the degree to which a granule contains X but not whether it contains X , so the criteria is lowered and more results are mined out. Furthermore, how strictly to evaluate G_g containing X can be adjusted by γ , which also effects the number of the final results. In summary, the extra FIs in FI-GF are caused by those methods which face the uncertainty brought by the granularity and ensure the appearance of the real FIs.

(4) Finally, when we check those results generated by FI-GF under $\gamma = 0.9$, $\gamma = 0.95$, and $\gamma = 1$, we can find that the higher γ is, the less results are mined out. Because γ is used to control how strictly to evaluate a granule covering an item, the higher γ is, the less results can satisfy this criteria, and vice versa.

Experiment 3. In Experiment 3, the granularity, Algorithm 1, in FI-GF is compared with a classical and widely applied progressive sampling algorithm, RC-SS [10], to prove its advantage.

RC-SS keeps increasing sample size until the similarity between two consecutive samples approaches to a high level. The similarity is evaluated through a representative set. A representative set is constructed by some item sets, and every item set in it contains the most frequent FI, which is mined by Apriori from this sample. Given 2 representative sets R_1 and R_2 , which are mined out from two samples S_1 and S_2 , if R_1 and R_2 are similar, S_1 and S_2 can also be regarded as similar. The similarity between two representative sets can be simply computed as follows:

$$\text{Sm}(R_1, R_2) = \frac{\sum_{x \in R_1 \cap R_2} \max(0, 1 - |\text{sup}_1(x) - \text{sup}_2(x)|)}{|R_1 \cup R_2|}, \quad (25)$$

where $\text{sup}_1(x)$ is the support of item set x , and it is computed based on the sample S_1 . RC-SS also needs a minimum support, which is denoted by $\min_sup$. This minimum support is used by Apriori to generate representative sets. Moreover, a sampling step, which decides the growth rate of sample size, is also needed, and a lowest of sample, which ensures that the final sample size will not be too small, is required too.

In this experiment, the representative set of a sample S is defined as those FIs in S which contain the most frequent

TABLE 3: The frequent item sets mined by FI-GF and Apriori.

Dataset	min_sup	μ	Results of FI-GF									Results of Apriori		
			$\gamma = 0.9$			$\gamma = 0.95$			$\gamma = 1$					
			1	3	6	1	3	6	1	3	6	1	3	6
Kosarak	0.1	0.5	1	3	11	1	3	11	1	3	11	1	3	11
			1	6	11	1	6	11	1	6	11	1	6	11
			3	6	11	3	6	11	3	6	11	3	6	11
T10I4D100K	0.0007	0.88	354	368	529	368	529	766	368	529	829	368	529	829
			354	368	722	368	529	829						
			354	368	766	368	766	829						
			354	368	829	529	766	829						
			354	529	722									
			354	529	766									
			354	529	829									
			354	722	766									
			354	722	829									
			354	766	829									
			368	529	722									
			368	529	766									
			368	529	829									
			368	722	766									
			368	722	829									
			368	766	829									
			529	722	766									
			529	722	829									
			529	766	829									
			722	766	829									
Retail	0.07	0.1	32	38	39	32	38	39	32	38	39	39	41	48
			32	38	41	32	38	41	32	38	41			
			32	38	48	32	38	48	32	38	48			
			32	39	41	32	39	41	32	39	41			
			32	39	48	32	39	48	32	39	48			
			32	41	48	32	41	48	32	41	48			
			38	39	41	38	39	41	38	39	41			
			38	39	48	38	39	48	38	39	48			
			38	41	48	38	41	48	38	41	48			
			39	41	48	39	41	48	39	41	48			
Connect	0.99	1	55	75	91	75	91	109	91	109	127	91	109	127
			55	75	109	75	91	127						
			55	75	127	75	109	127						
			55	91	109	91	109	127						
			55	91	127									
			55	109	127									
			75	91	109									
			75	91	127									
			75	109	127									
			91	109	127									
Mushroom	0.6	1	34	36	85	34	36	85	34	85	86	34	85	86
			34	36	86	34	36	86	34	85	90	34	85	90
			34	36	90	34	36	90	34	86	90	34	86	90
			34	85	86	34	85	86	85	86	90	85	86	90
			34	85	90	34	85	90						
			34	86	90	34	86	90						
			36	85	86	36	85	86						
			36	85	90	36	85	90						
			36	86	90	36	86	90						
			85	86	90	85	86	90

FIGURE 9: The time consumed by RC-SS and the granularity when they are applied to the datasets in Table 1.

item, the min_sups of RC-SS are all set to 0.1, the sampling steps are all set to 100, and the lowest sample sizes are set to 300. If 6 latest and consecutive $Sm(\mathbf{R}_1, \mathbf{R}_2)$ are all higher than 0.9, the sampling will be stopped. The parameters of the granularity are the same as Table 2.

Figure 9 draws the time which is, respectively, consumed by the granularity and RC-SS. Figure 10 shows the transaction scales generated by RC-SS and granule scales generated by the granularity. Figure 11 shows the learning curves of the RC-SS, which describes the changes of $Sm(\mathbf{R}_1, \mathbf{R}_2)$ when RC-SS is applied to those datasets in Table 1.

Through Figures 9–11, the following can be got.

(1) According to Figure 9, we can know that, in most cases, the time cost by the granularity is less than the time cost by RC-SS. This phenomenon can be explained through the following. To begin with, RC-SS has to repeatedly generate samples until the similarity between two consecutive samples becomes stable and high. Secondly, RC-SS has to apply Apriori to mine the representative sets out from every sample, which means that the sampling process still costs too much time to scan the transactions. Furthermore, the complexity of (25), which is the computation of similarity, also puts some burden on RC-SS.

(2) However, according to Figure 9, we can know that when the granularity and RC-SS are applied to the dataset retail, the time cost by the granularity is more than the time cost by RC-SS, and when the granularity and RC-SS are applied to the dataset kosarak, the granularity only has a slight advantage on efficiency. This phenomenon is caused by how FI-GF builds its granules. When FI-GF is building a granule, a collection of transactions needs to be built firstly. To ensure that the information in a collection can be represented more precisely, differences in the collection are computed by (6), and (6) needs to compute the hamming distance

FIGURE 10: The number of transactions sampled by RC-SS and granules generated by the granularity.

between two transactions. If the lengths of two transactions are longer, the computing of hamming distance will be more complex. Thus, considering the average transaction lengths of retail and kosarak are both very long, computations of hamming distances on them will certainly become slower, and meanwhile, the granularity takes more time.

(3) According to Figure 10, we can know that, in most cases, the granule scale which is built by the granularity is less than the transaction scale which is chosen by RC-SS. This phenomenon can be explained as follows. As mentioned in Section 3.1, the average transaction scale represented by a granule can be controlled by α and β , and the more transactions a single granule can represent, the less granules will be generated. Therefore, if α and β are set reasonably, the granule scale can be controlled in a low level, and this is also an advantage of the granularity. The granularity not only can make the simplified dataset as precise as possible but also provides a tool to control the granule scale. However, the progressive sampling can only do the first job, and the final sample size cannot be controlled. For example, through Figure 10, we can know that, even after the sampling, the final sample size of the dataset connect still approaches to 10300, which will also cost too much time for scanning.

(4) According to Figure 11, we can know that the learning curves of RC-SS are not always smooth and convergent. When RC-SS is applied to mine the datasets kosarak, retail, and mushroom, the convergences of learning curves are good. However, when RC-SS is applied to mine the datasets T10I4D100K and connect, RC-SS did not perform well. The reason of this phenomenon can be explained as follows.

The stop condition of RC-SS in this experiment is that 6 consecutive $Sm(\mathbf{R}_1, \mathbf{R}_2)$ are all higher than 0.9. The more elements $\mathbf{R}_1 \cap \mathbf{R}_2$ has, the higher $Sm(\mathbf{R}_1, \mathbf{R}_2)$ will be. Meanwhile, the representative set of a sample is constructed by

FIGURE 11: The self-similarity of RC-SS when it is applied to the datasets in Table 1.

those FIs covering the most frequent item in this sample. If several items have the similar and high frequencies, a little change of the sample may change the rank of frequency and may result in a big change of the representative sets as well, so the similarity between the original sample and the sample after the tiny change is low. Thus, the larger the variance of the frequencies of items is, the better the convergence of RC-SS will be. Therefore, the different variances of the frequencies of items in the datasets in Table 1 cause the different convergences. This characteristic shows another disadvantage of RC-SS when compared with FI-GF, which is that RC-SS cannot adapt to any dataset well.

5. Conclusions

The following conclusions can be deduced:

- (1) FI-GF has both the high efficiency and reliability. Firstly, the granularity of it not only decreases the scale of transactions but also ensures the precision of every granule. Then, the support computed by fuzzy theory successfully adapts to the uncertainty brought by the granularity.
- (2) In most of time, compared with RC-SS, which is the most representative progressive sampling way,

the granularity costs less time. Furthermore, the granularity can control the final size of simplified dataset to some extent and can adapt to more datasets.

- (3) The application of granular computing to FI mining has broad prospects. The next work is to solve the problem of the huge computation brought by the long transactions.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] J. Han, M. Kamber, and J. Pei, *Data Mining Concepts and Techniques*, China Machine Press, Beijing, China, 3rd edition, 2012.
- [2] K. Luo, L.-L. Wang, and X.-J. Tong, "Mining association rules in incomplete information systems," *Journal of Central South University of Technology (English Edition)*, vol. 15, no. 5, pp. 733–737, 2008.
- [3] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules," in *Proceedings of the International Conference on Very Large Data Bases*, pp. 487–499, Santiago de Chile, Chile, 1994.
- [4] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '00)*, pp. 1–12, ACM, Dallas, Tex, USA, May 2000.
- [5] M. J. Zaki, "Scalable algorithms for association mining," *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 3, pp. 372–390, 2000.
- [6] A. Pietracaprina, M. Riondato, E. Upfal, and F. Vandin, "Mining top- K frequent itemsets through progressive sampling," *Data Mining and Knowledge Discovery*, vol. 21, no. 2, pp. 310–326, 2010.
- [7] M. J. Zaki, S. Parthasarathy, W. Li, and M. Ogihara, "Evaluation of sampling for data mining of association rules," in *Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE '97)*, pp. 42–50, IEEE, Birmingham, UK, April 1997.
- [8] V. T. Chakaravarthy, V. Pandit, and Y. Sabharwal, "Analysis of sampling techniques for association rule mining," in *Proceedings of the 12th International Conference on Database Theory (ICDT '09)*, pp. 276–283, ACM, Saint-Petersburg, Russia, March 2009.
- [9] H. Toivonen, "Sampling large databases for association rules," in *Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB '96)*, pp. 134–145, Mumbai, India, September 1996.
- [10] S. Parthasarathy, "Efficient progressive sampling for association rules," in *Proceedings of the 2nd IEEE International Conference on Data Mining (ICDM '02)*, pp. 354–361, Maebashi, Japan, December 2002.
- [11] B. Chen, P. Haas, and P. Scheuermann, "A new two-phase sampling based algorithm for discovering association rules," in *Proceedings of the International Conference on Knowledge Discovery in Databases (KDD '02)*, pp. 462–468, 2002.
- [12] Y. Y. Yao, "Granular computing: basic issues and possible solutions," in *Proceedings of the 5th Joint Conference on Information Sciences (JCIS '00)*, pp. 186–189, Atlantic City, NJ, USA, February–March 2000.
- [13] W. Pedrycz, *Granular Computing: Analysis and Design of Intelligent Systems*, CRC Press, London, UK, 1st edition, 2013.
- [14] W. Pedrycz and W. Homenda, "Building the fundamentals of granular computing: a principle of justifiable granularity," *Applied Soft Computing*, vol. 13, no. 10, pp. 4209–4218, 2013.
- [15] L. A. Zadeh, "Fuzzy sets," *Information and Computation*, vol. 8, pp. 338–353, 1965.
- [16] D. Dubois, E. Hüllermeier, and H. Prade, "A systematic approach to the assessment of fuzzy association rules," *Data Mining and Knowledge Discovery*, vol. 13, no. 2, pp. 167–192, 2006.
- [17] T.-P. Hong, K.-Y. Lin, and B.-C. Chien, "Mining fuzzy multiple-level association rules from quantitative data," *Applied Intelligence*, vol. 18, no. 1, pp. 79–90, 2003.
- [18] B. Pei, S. Zhao, H. Chen, X. Zhou, and D. Chen, "FRAP: mining fuzzy association rules from a probabilistic quantitative database," *Information Sciences*, vol. 237, pp. 242–260, 2013.
- [19] M. Delgado, N. Marín, D. Sánchez, and M.-A. Vila, "Fuzzy association rules: general model and applications," *IEEE Transactions on Fuzzy Systems*, vol. 11, no. 2, pp. 214–225, 2003.
- [20] A. Salam and M. S. H. Khayal, "Mining top- k frequent patterns without minimum support threshold," *Knowledge and Information Systems*, vol. 30, no. 1, pp. 57–86, 2012.
- [21] P. Tzvetkov, X. Yan, and J. Han, "TSP: mining top- k closed sequential patterns," *Knowledge and Information Systems*, vol. 7, no. 4, pp. 438–457, 2005.
- [22] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 207–216, ACM, May 1993.
- [23] W. Pedrycz, "Granular computing-the emerging paradigm," *Journal of Uncertain Systems*, vol. 1, no. 1, pp. 38–61, 2007.
- [24] W. Pedrycz, "From numeric models to granular system modeling," *Fuzzy Information and Engineering*, vol. 7, no. 1, pp. 1–13, 2015.
- [25] A. Bargiela and W. Pedrycz, "The roots of granular computing," in *Proceedings of the IEEE International Conference on Granular Computing*, pp. 806–809, Atlanta, Ga, USA, May 2006.
- [26] P. Z. Wang, "From the fuzzy statistics to the falling random subsets," in *Advances in Fuzzy Sets, Possibility Theory, and Applications*, pp. 81–96, Springer, New York, NY, USA, 1983.