An Efficient and Adaptive Granular-Ball Generation Method in Classification Problem

Shuyin Xia, *Member, IEEE*, Xiaochuan Dai, Guoyin Wang, *Senior Member, IEEE*, Xinbo Gao, *Senior Member, IEEE*, and Elisabeth Giem

***Abstract*— Granular-ball computing (GBC) is an efficient, robust, and scalable learning method for granular computing. The granular ball (GB) generation method is based on GB computing. This article proposes a method for accelerating GB generation using division to replace *k*-means. It can significantly improve the efficiency of GB generation while ensuring an accuracy similar to that of the existing methods. In addition, a new adaptive method for GB generation is proposed by considering the elimination of the GB overlap and other factors. This makes the GB generation process parameter-free and completely adaptive in the true sense. In addition, this study first provides mathematical models for the GB covering. The experimental results on some real datasets demonstrate that the two proposed GB generation methods have accuracies similar to those of the existing method in most cases, while adaptiveness or acceleration is realized. All the codes were released in the open-source GBC library at http://www.cquptshuyinxia.com/GBC.html or https://github.com/syxiaa/gbc**

***Index Terms*— Class noise, granular ball (GB), granular-ball computing (GBC), granular computing, GB generation, label noise.**

I. Introduction

COGNITIVE computing combined with human cognitive mechanisms makes the decision-making process more reliable, efficient, and understandable. It is an important means to achieve reliable governance of information spaces and is an important direction for the development of artificial intelligence. Chen's [1] research results, published in the journal *Science* in 1982, revealed that human cognition is characterized by large-scale priority. Large outline letters are seen first, followed by small letters within the outline letters, as shown in Fig. 1. Granular computing can achieve efficient, scalable, and robust learning based on this cognitive characteristic. Zadeh [2], [3] proposed the problem of granular information granulation and the concept of granular computing. After decades of continuous research by scholars at home and abroad, fuzzy sets [4], [5], [6], [7] and rough sets [8], [9], [10], [11] have been developed. Zhang and Zhang [12], [13] proposed the quotient space theory, Li *et al.* [14], [15] proposed the cloud model theory, and Xia *et al.* [16], [17] developed the granular-ball computing (GBC) as well as other model methods.

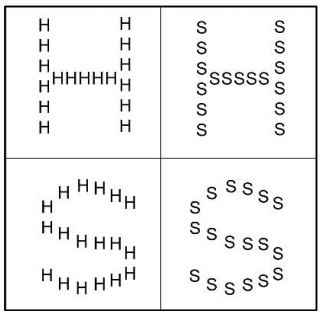


Fig. 1. Human cognition–the coarse-grained large range is preferred.

Data elements in fuzzy sets are described by the membership degree of different granularities of fuzzy information. The rough set theory and quotient space theory use equivalence relations and equivalence classes to construct granules of different sizes. The universe of rough set theory is the point set of objects, and topological relations between elements are not considered. The quotient space theory is studied under the condition that topological relations exist between the elements in the universe. Quotient space and cloud models are two important granular computing methods. Among these, rough sets have been the most widely studied, and many scholars have made significant efforts in the field of rough sets. Wang [18] discovered a distinction between a rough set's algebraic and information entropy forms. Pei and Miao [19] discovered equivalence between fuzzy soft sets and fuzzy information systems. Qian *et al.* [10] used positive field reduction and nuclear attributes to reduce the number of

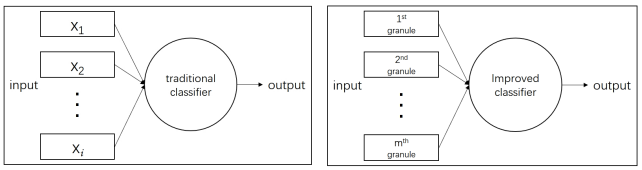
Manuscript received 31 December 2021; revised 14 June 2022; accepted 22 August 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2019QY(Y)0301; in part by the National Natural Science Foundation of China under Grant 62222601, Grant 62176033, Grant 61936001, and Grant 62221005; in part by the Key Cooperation Project of the Chongqing Municipal Education Commission under Grant HZ2021008; in part by the Natural Science Foundation of Chongqing under Grant cstc2019jcyj-cxttX0002; and in part by the National Science Foundation of U.S. under Award 1631776. *(Corresponding author: Guoyin Wang.)*

Shuyin Xia, Xiaochuan Dai, Guoyin Wang, and Xinbo Gao are with the Chongqing Key Laboratory of Computational Intelligence, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: xiasy@cqupt.edu.cn; daixiaochuaner@qq.com; wanggy@cqupt.edu.cn; gaoxb@cqupt.edu.cn).

Elisabeth Giem is with the Department of Computer Science and Engineering, University of California at Riverside, Riverside, CA 92521 USA (e-mail: gieme01@ucr.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2022.3203381.

Digital Object Identifier 10.1109/TNNLS.2022.3203381



(a) (b)

Fig. 2. Comparison between multi granularity classifier and traditional classifier. (a) Traditional classifier method. (b) Classifier method of coarse-grained input.

calculations in the positive field of an attribute combination explosion. Xu and Wang [20] established the topological structure of a covering rough-set model. Yao [21] pointed out that the label noise has an obvious interference effect on the upper and lower approximate calculations. Hu *et al.* [22] designed a robust classification algorithm based on fuzzy lower approximation and considered the data distribution information and was included in the calculation and fuzzy approximation of the data distribution of rough set model perception [23]. Wu and Leung [24] proposed a multiscale decision table. Chen *et al.* [25] proposed a dynamic incremental approximation method based on a rough-set maintenance environment. Xia *et al.* [8] proposed a parameterless rough set method that can process continuous data without relying on the membership function.

In granular computing, the larger the granularity, the higher the efficiency, and the better the robustness to noise; however, it is also more likely to cause neglect of details and loss of accuracy. The smaller the granularity, the more attention is paid to the details, but this may reduce efficiency and deteriorate robustness to noise. Selecting different granularities according to different scenes can improve the performance of the multigranularity learning method. Although multigranular computing has a long research history, as cognitive computing science, it also faces some new challenges and needs new development. For example, in terms of the "classifier," one of the most widely used methods of artificial intelligence, as shown in Fig. 2(a), the inputs of most existing classifiers are the finest-grained sample points or pixels [26], [27], [28]; therefore, coarse-grained characterization is lacking. Some researchers have proposed classification algorithms based on multigranularity ideas. For example, Dick and Kandel [29] assumed that the connection weights are all linguistic variables and that they have different granulation of connection weights. Weight updating is realized by adding "linguistic hedges," but this method sacrifices accuracy. Leite *et al.* [30] used fuzzy neurons to build interpretable multisize local models [31], [32], which can learn fuzzy rules, and the output space can be processed as membership information, which can be used to process fuzzy data. In a few cases of machine learning and data mining, such as in [33], fuzzy rules are extracted for credit card fraud detection. The purpose of these studies is to use neural networks to process fuzzy data and apply them to fuzzy control. It is not based on multigranularity ideas to improve the scalability, efficiency, or robustness of the classifier, and its essence is still a point-input method. Park *et al.* [34] constructed the information granule by feature selection in the input space, which is essentially a feature preprocessing method without changing the learning mode of the neural networks. Tang and He [35] introduced a method of sampling and mapping information granules in a support vector machine. The research work of [34] and [35] focuses more on understanding existing research using the concept of multigranularity. Pedrycz and Vukovich [36] systematically proposed a neural network granulation framework from the input layer and output layers. Both the rough set and fuzzy methods can granulate the input space. However, this study did not realize a specific multigranularity neural network and examined its performance advantages. As shown in Fig. 2(a), the input of the traditional classifier is composed of the finest-grained granules, i.e., data points. In the multigranularity classifier in Fig. 2(b), the input are no longer the finest-grained points but some abstract granules. By adjusting the input granules, so that the improved classifier becomes a multigranularity classifier. The design of this granularity should meet high-dimensional scalability, i.e., no complicated calculations are required in the high-dimensional space. For this reason, Xia *et al.* [17] proposed a granular ball (GB) computing method using ball as "granule." This is because the geometry of a ball is completely symmetrical, and only two data points are needed to characterize it in any dimension: center and radius, so it is convenient to apply to high-dimensional data. At the same time, they also proposed an efficient and adaptive method to generate GBs.

Furthermore, GB computing is introduced into the classifier, the framework of the GB computing classifier is proposed, the original model of the GB support vector machine (GBSVM) is derived, and the k-nearest neighbor algorithm of the GB (GBkNN) is proposed [17]. The efficiency of the GBkNN is hundreds of times higher than that of the existing k-nearest neighbor (kNN) algorithm, especially for large-scale data. In addition, the GBkNN does not need to select parameter k and helps to alleviate the performance in unbalanced data, which is not available in existing kNN algorithms. Owing to the robustness of GB computing, the GBkNN has higher accuracy than the accurate kNN in many datasets. In addition, GB computing was introduced into the neighborhood rough set (NRS), and a new rough set method called "GBs neighborhood rough set (GBNRS)" was developed [8]. GBNRS is the first parameter-free rough-set algorithm to process continuous data without prior knowledge (i.e., setting the membership function), which is more efficient than NRS. Because GBNRS can adaptively select the neighborhood radius, it can obtain a higher classification accuracy than NRS in many cases. In addition, the GB computing was introduced into the k-means algorithm and a simple and fast k-means clustering method "ball k-means" was developed [16]. Ball k-means is dozens of times more efficient than similar algorithms, especially in the challenging large-k clustering problem. GB computing is efficient, robust, and scalable [17]. However, there are still many challenges in GB generation, such as the optimization of the purity threshold and its efficiency improvement. The main contributions of this study are as follows.

1) The acceleration GB generation method is proposed using the division to replace k-means. It can accelerate

the GB generation several times to dozens of times while a similar accuracy is achieved.

2) A new adaptive method for the GB generation is proposed by considering the GB's overlap elimination and some other factors. This makes the GB generation process parameter-free and completely adaptive in the true sense.
3) This article first provides the mathematical models for the GB covering.
4) A fully adaptive GBkNN algorithm is indirectly proposed based on the adaptive GB generation method.

II. Related Work

A. GB Computing

Combining the theoretical basis of traditional granular computing, and based on the research results published by Chen [1] in *Science* in 1982, he pointed out that "human cognition has the characteristics of large-scale priority," and Wang [37] put forward much granular cognitive computing. Based on granular cognitive computing, GB computing is a new, efficient and robust granular computing method proposed by Xia *et al.* [17], the core idea of which is to use "GBs" to cover or partially cover the sample space. A GB $= \{x_i(i = 1, \ldots, n)\}$, where x_i represents the objects in GB and N is the number of objects in GB. GB's center C and radius r are represented as follows:

$$C = \frac{1}{N}\sum_{i=1}^{N} \boldsymbol{x_i} \tag{1}$$

$$r = \frac{1}{N}\sum_{i=1}^{N} |x_i - C|. \tag{2}$$

This implies that the radius is equal to the average distance from all objects in GB to its center. The radius can also be set as the maximum distance. The "GB" with a center and radius is used as the input of the learning method or as accurate measurements to represent the sample space, achieving multigranularity learning characteristics (i.e., scalability, multiple scales, etc.) and the accurate characterization of the sample space. In addition, we give the definition of purity in Definition 1.

Definition 1: Given a GB, which is composed of k classes of samples, i.e., that $\bigcup_{i=1}^{k} c_i = \text{GB}$ and $\bigcap_{i=1}^{k} c_i = \emptyset$, where c_i represents the ith class of samples in GB. The purity T_{GB} of GB can be expressed as

$$T_{\text{GB}} = \frac{\max|\boldsymbol{c_i}|}{|\text{GB}|}, \quad i = 1, 2, \ldots, k \tag{3}$$

where $|.|$ represents the number of samples in a set.

The generation of GBs requires that the purity of GBs does not meet the given purity threshold. If the purity of the GB meets the purity threshold, the splitting is stopped. The basic process of GB generation for classification problems in GB computing is shown in Fig. 3.

As shown in Fig. 3, to simulate the "characteristics of the large-scale priority of human cognition" at the beginning of the algorithm, the entire dataset can be regarded as a GB.

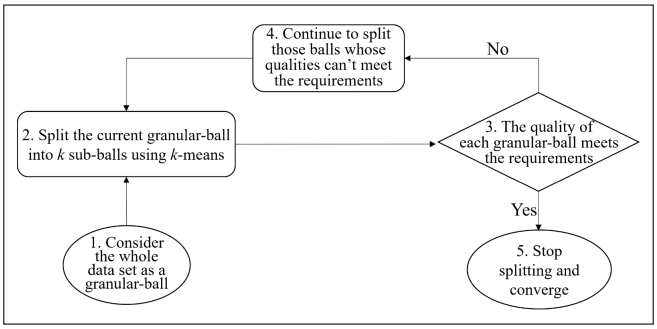


Fig. 3. Process of the existing GB generation in GB computing.

At this time, the purity of the GB was the worst and could not describe the distribution characteristics of the data. The "purity" is used to measure the quality of a GB [17] in step 3 in Fig. 3. This was equal to the proportion of most labels in the GB. Then, the number of different classes in the GB is counted and denoted as m; the GB is split into m child GBs in step 2. In step 3, the purity of each GB is calculated; if a GB does not reach the purity threshold, it must be split. As the splitting process continues to advance, the purity of the GBs increases, the decision boundary becomes increasingly clearer, the boundary is clearest, and the algorithm converges when the purity of all GBs meets the requirements. It can be concluded from [17] that for a dataset, regardless of the distribution of its data, we can describe its decision boundary using sufficient GBs.

An example of GB generation on the dataset fourclass is shown in Fig. 4. At the beginning of the algorithm, as shown in Fig. 4(a), the entire dataset can be considered as a GB. As the fourclass contains two classes of points, k in step 2 in Fig. 3 is equal to 2, and two heterogeneous points are randomly selected as the initial centers of the two-child GBs. The experimental results are presented in Fig. 4(b). However, the GBs are too coarse, and the qualities of the GBs are not sufficiently high, that is, their purities do not reach the purity threshold. Therefore, the decision boundary of the GBs was inconsistent with that of the dataset. As the splitting process progresses, as shown in Fig. 4(c)–(d), the GBs become finer, and the purity of each GB becomes high until it reaches the purity threshold or other quality measurements. As shown in Fig. 4(e), each GB reached the purity threshold and was sufficiently fine. At this time, the decision boundary was consistent with that of the dataset. Fig. 4(f) shows the extracted GBs when the points are removed.

The more the number of GBs, the higher the purity of the whole GBs, and the more stable the splitting result. If the number of GBs is too small, it means that some GBs have poor quality and can easily generate higher-quality GBs, and the splitting result is not sufficiently stable. As shown in Fig. 4(b), the number of GBs is very small. At this time, the GBs cannot perfectly reflect the data distribution and can be easily divided into high-quality GBs. If the number of GBs is the same as the number of points in the dataset, the purity of each GB is one. As shown in Fig. 4(e), the number of GBs is very large, which perfectly reflects the data distribution, and it is difficult

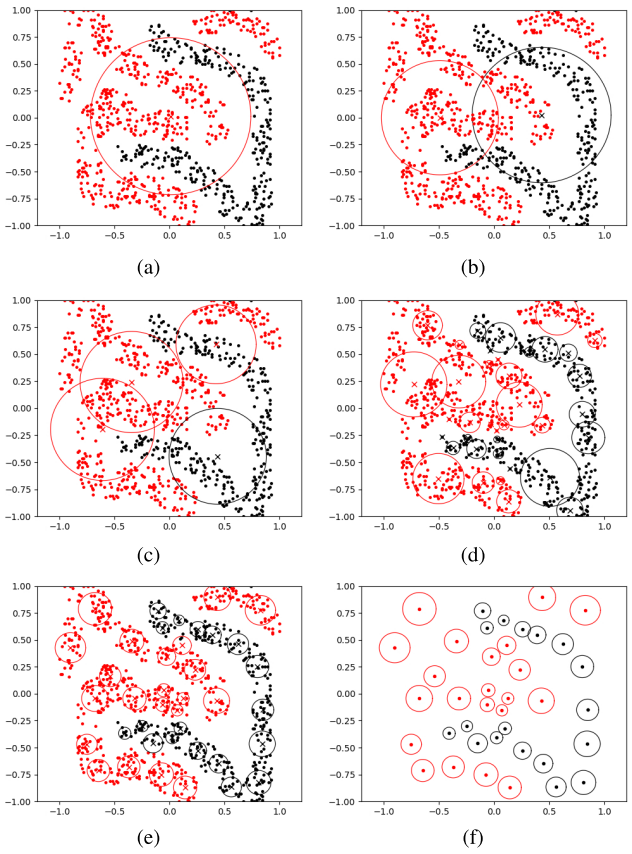


Fig. 4. GB splitting generation process of the existing method on the dataset fourclass. The colors of the two GBs in the figure (corresponding to the two-sample point colors), respectively, represent the two types of category labels. (a) Initial GB, the whole dataset can be seen as a GB to participate in subsequent iterations. (b) GBs generated in the first iteration. (c) GBs generated in the second iteration. (d) Stop splitting results. (e) Results after stopping splitting. (f) GBs extracted.

to generate higher-quality GBs. At this point, the splitting result is very stable.

GB computing has led to the development of GB classifiers [17], GB clustering [16], GBNRS [8], and GB sampling (GBS) methods [38].

B. Granular-Ball kNN

The kNN algorithm has many characteristics, such as simplicity and natural response to multiple classifications, independent of training, can be used for classification and regression at the same time, and can be easily to parallelize and implemented. This is one of the most widely used artificial intelligence algorithms. In kNN, the basic principle of finding the kNNs of a query point is to compute the Euclidean distance from the query point of all data points and use the values (or labels) of these neighbors to predict or classify the query point. This method is called a full-search algorithm (FSA). The FSA has the following common problems: it needs to optimize the value of k, and the optimization of the value of k requires quadratic time complexity, so it is very time-consuming. From the perspective of multigranularity, the source of the problem of k optimization is excessive fine-grained attention. For this reason, in [17], we introduce a GB into kNN and propose an efficient nearest neighbor algorithm without optimizing k value, GBkNN. The basic idea is very easy to implement. Based on GB computing, a single query sample point is a GB with a very small radius, and its predicted label is equal to the nearest GB's label, which is determined by the label of most samples in the GB. Therefore, the common feature of GBkNN and traditional kNN is that the mark of the query point is determined by many points. However, the main advantage of GBkNN is that there is no need to optimize the parameter k, and the query point label is determined by the adaptively generated nearest neighbor GB with different coarse grains; the second advantage of GBkNN is that the number of GBs is much smaller than the sample points, and the calculation amount of the nearest GB queried by the query point is much less than kNN, resulting in higher efficiency of GBkNN than traditional kNN; the third advantage is that the decision of traditional kNN can be affected by label noise, but GBkNN regards some minority samples as noise, so that it have good robustness, especially on noisy datasets. These three points are important advantages of the GBkNN.

C. GB Sampling

The purpose of GBS is to decrease the size of a dataset in classification by introducing the concept of GB computing. The GBS method uses adaptively generated balls to cover the data space, and the points near the boundary of each GB constitute the sampled results [38]. Fig. 5 shows the basic idea of GBS. Fig. 5(a) shows the original dataset and its decision boundary. In Fig. 5(b), we find the intersections of the coordinate axis with the GB center as the origin and GB. In a GB, the points closest to these intersections constitute the sampled result among the points with the same label as the ball. These points were located near the boundaries of the GB. The same label can filter the effect of label noise points. For example, for the GB A in Fig. 5(b), the intersection points of the ball and the coordinate axis with the center of the ball as the origin are a–d, and the points with the same label as the ball closest to these intersections are a'–d' in A. Therefore, a'–d' are the sampling results in A, and they are also the best points for describing the boundary of the GB A. Therefore, as shown in Fig. 5(c), the boundary generated by GBS is closer to the boundary of the original dataset than the boundary generated by random sampling.

For the intersection point a on the GB, it can be expressed as the point where the center point vector c moves the length r along the specified coordinate axis. The moving direction of the center vector includes positive and negative, so the coordinate axis corresponds to two intersection points. Specifically, for a d-dimensional dataset D, the center point vector c of the ith GB generated on D is $c = (c_i^1, c_i^2, \ldots, c_i^j, \ldots, c_i^d)$, and radius r_i. The two intersection points in the positive and negative directions of the jth coordinate axis can be expressed as follows:

$$b_j^+ = \left(c_i^1, c_i^2, \ldots, c_i^j + r_i, \ldots, c_i^d\right) \tag{4}$$

$$b_j^- = \left(c_i^1, c_i^2, \ldots, c_i^j - r_i, \ldots, c_i^d\right). \tag{5}$$

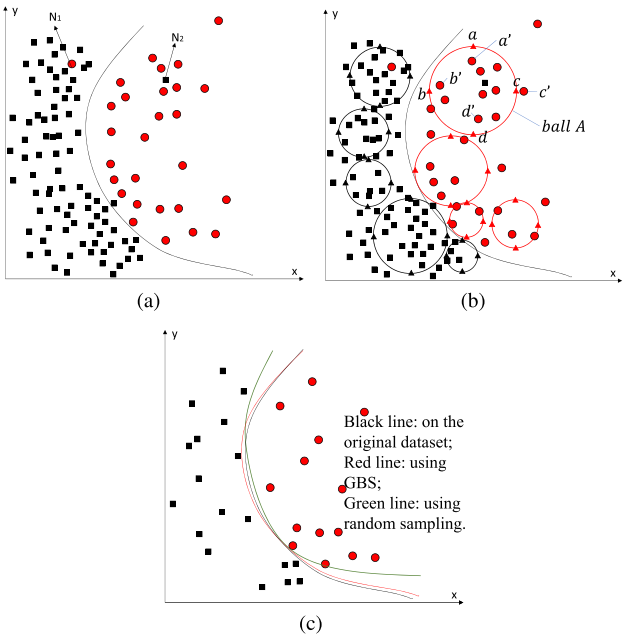


Fig. 5. Schematic of GBS. (a) Original dataset and its boundaries. (b) $2 \times d$ intersection points a–d. The points a'–d' are the closest points to the intersection points in the GB A. (c) Sampling results are obtained from random sampling [38].

In the balanced datasets, the sampling point set S_k in the kth GB can be expressed as

$$S_k = \begin{cases} \{x_i | x_i \in \mathrm{GB}_k, \mathrm{label}(x_i) = \mathrm{label}(\mathrm{GB}_k)\} & |\mathrm{GB}_k| \leq 2d \\ \{x_i | x_i \in S_{km}\} & |\mathrm{GB}_k| > 2d \end{cases} \tag{6}$$

where

$$\begin{aligned} S_{km} = &\{x_i | \min\{\mathrm{dis}(x_i, b_m^+)\}, x_i \in \mathrm{GB}_k, \mathrm{label}(x_i) = \mathrm{label}(\mathrm{GB}_k)\} \\ &\cup \{x_i | \min\{\mathrm{dis}(x_i, b_m^-)\}, x_i \in \mathrm{GB}_k, \mathrm{label}(x_i) = \mathrm{label}(\mathrm{GB}_k)\} \end{aligned} \tag{7}$$

Equation (6) indicates that the labels of the points in S_k are consistent with the kth GB label. Two points were sampled for each dimension. So, in (7), S_{km} is used to express a set that includes two sampling points in the mth dimension of the kth GB. For unbalanced datasets, $S_k \in \mathrm{GB}_k$ needs to be changed to $S_k \in \mathrm{majGB}_k$, majGB_k represents the majority of the GBs.

In classification with label noise, GBS can reduce a dataset and improve its data quality. GBS is also effective for undersampling unbalanced classifications. In addition, the time complexity of GBS is $O(n)$; therefore, it can speed up most classifiers [39].

III. GB Covering Model

At present, the GB covering of GB computing lacks a mathematical model. In this section, we establish the basic model of the GB covering, which is described as follows: given a dataset $D = \{x_i (i = 1, \ldots, n)\}$, where n is the number of samples on D. GBs $\mathrm{GB}_1, \mathrm{GB}_2, \ldots, \mathrm{GB}_m$ are used to cover and represent the dataset D. The original goal of the optimization problem of the GB generation method is expressed as OO_{bj}, and the main factors for measuring the coverage are as follows.

1) When other factors remain unchanged, the higher the coverage, the less the sample information is lost, and the characterization is more accurate. Suppose the number of samples in the jth GB GB_j is expressed as $|\mathrm{GB}_j|$; then, its coverage degree can be expressed as $\sum_{j=1}^{m}(|\mathrm{GB}_j|)/n$.
2) When other factors remain unchanged, the number of GBs is related to their size. The fewer the number of GBs, the coarser the GBs, and the more coarse the granularity characteristics: the more efficient the GB calculation, the better the robustness.
3) In addition, under different problems, to correspond to the relevant optimization goal OO_{bj}, the quality of the GB GB_j must be higher than a given purity threshold T of the given evaluation method.

This factor is also related to the lower limit of the size of the GBs, so that the GBs must be "fine" enough to accurately describe the problem. The threshold can be obtained in a given manner, in a lattice search, or in an adaptive manner, which we pursue. Taking the reciprocal of the GB covering to optimize its minimum value, the optimization goal of the GBs can be expressed as

$$\begin{aligned} &\mathrm{Min}\ \lambda_1 * n / \sum_{j=1}^{m}\left(|\mathrm{GB}_j|\right) + \lambda_2 * m \\ &\text{s.t.}\ \ \mathrm{quality}\left(\mathrm{GB}_j\right) \geq T \end{aligned} \tag{8}$$

where λ_1 and λ_2 are the corresponding weight coefficients and $m < n$. The minimum number of GBs should be considered to obtain the maximum coverage degree when generating GBs. In this equation, the coverage degree should be guaranteed and the number of GBs should be as small as possible. By adjusting the parameters λ_1 and λ_2, the optimal GB generation results can be obtained to minimize the value of the whole equation.

The definition of a GB's quality differs according to the environment, but it can be defined as a sample label with a certain approximate (or equivalence) relation. For example, in the classification problem, we often use the nearest neighbor to describe this equivalence relation. Thus, these factors are crucial. It is unreasonable to rely only on factor 1, coverage degree, or factor 2, the number of GBs. For example, in the extreme case shown in Fig. 6(a), only one GB was used. At this time, the quality of the GB is poor, and one GB cannot describe the distribution of a dataset (i.e., data boundary). If factor 2 is not considered, as shown in Fig. 6(b), the GBs can only cover a small part of the dataset, and it is impossible to describe the dataset. If do not consider the factor 3 "the number of GBs" (i.e., the size of the GBs), and only consider the quality and coverage degree, the GBs can be divided into the finest GB, i.e., a GB contains only one sample point. Coarse granularity does not make sense. Therefore, none of the above factors is indispensable. Overall, when factor 1 ensures a certain level of coverage, factors 2 and 3 obtain GBs with appropriate GB size; when factors 1 and 2 remain constant, the smaller the threshold of factor 3, the

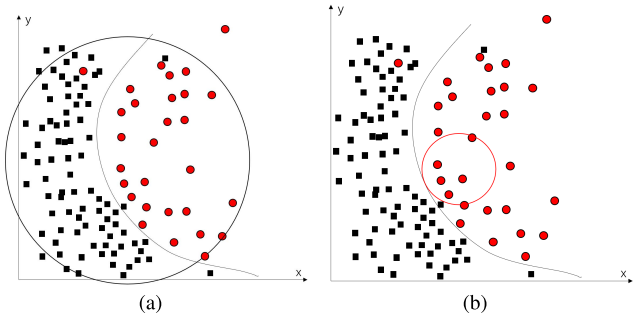


Fig. 6. Invalid coverage of sample space by GBs. GB coverage results without considering (a) quality of the GB and (b) rate of the coverage of the GBs.

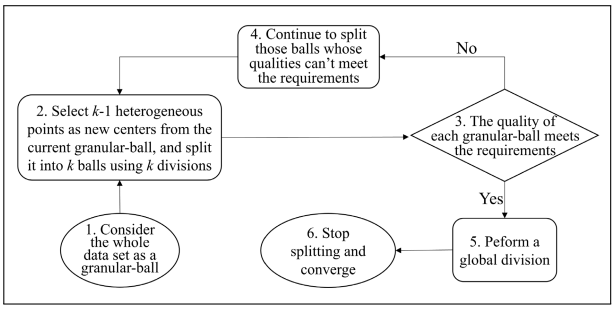


Fig. 7. Process of the acceleration GB generation in GB computing.

easier it is to satisfy the quality of the GBs (i.e., the coarser the GBs, the fewer the number of the GBs, and the more efficient computational performance can be obtained). The control of the threshold in Factor 3 exhibits the scalability ability of GB generation. The existing GB generation method, as shown in Fig. 3, provides a heuristic optimization strategy.

IV. Acceleration GB Generation Method

A. Motivation

The existing GB generation method uses the k-means algorithm to split the GB; thus, GB generation is not more efficient than k-means, which can generate stable splitting results in each iteration of GB generation. However, stability in the intermediate process is not required in the process; what is required is only to generate GBs fulfilling (8), such as that the lower bound should be ensured.

B. Process of the Acceleration GB Generation Method

As stability in the intermediate process is not needed, as shown in step 2 in Fig. 7, we use one division, that is, one iteration process in the k-means, to split a GB instead of an entire k-means algorithm. In addition, in contrast to the existing method shown in Fig. 3, a global division is added to the end to improve the whole distribution of the final GBs. In global division, the division is performed based on all division points. The specific process is illustrated in Fig. 7. To describe the process of the acceleration GB generation more clearly, we first define the father ball and the child ball.

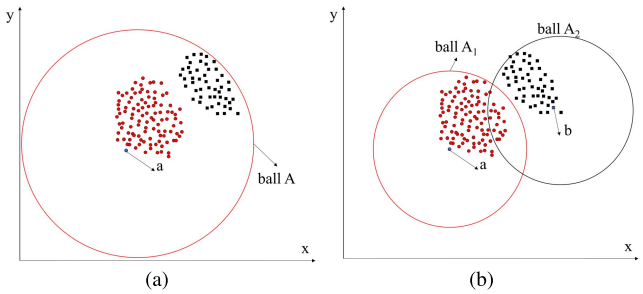


Fig. 8. GB splitting using the acceleration GB generation method. (a) GB A with center as a. (b) A is split into two child balls A_1 and A_2 whose centers are a and b, respectively.

Definition 2: Given A and $A_i (i = 1, 2, \ldots, k)$, we suppose that $\bigcup_{i=1}^{k} A_i = A$ and $\bigcap_{i=1}^{k} A_i = \emptyset$. A and A_i are the parent and child balls, respectively.

The k-means algorithm consists of t iterations, where t denotes the number of iterations, and each iteration is a division in which all points are divided into k clusters according to their distances to the k center points. In step 2 of Fig. 7, the k-means used for splitting a GB is replaced with k division, where k denotes the number of classes in a GB. Therefore, the computational cost is directly decreased. In addition, as shown in step 2, taking the splitting GB A as an example, the center point of the GB A, denoted as a, remains as the center point of a certain child ball of A; therefore, only $k-1$ points are selected as the centers of the new $k-1$ child balls of A. The sample points in all k child balls do not need to calculate the distance from the original center point a, because they have been calculated previously. Consequently, the computational cost further decreases.

As shown in Fig. 8, a GB A with a center as a is split into two child balls A_1 and A_2 whose centers are a and b, respectively. The radius of a GB is represented by the furthest distance from the data points to its center to cover all the data points in the ball. The center a of ball A in Fig. 8(a) remains at the center of child ball A_1 in Fig. 8(b). In this split process of the GB, the distance from all data points in the ball A to the center a does not need to be computed again, and only the distance from all the data points to the center b of the child ball A_2 in Fig. 8(b) is computed. Finally, all the data points are divided into two GBs based on the distances above the two centers a and b.

In the GB splitting process of the acceleration method, the center of the GB is its division point instead of that computed using (1). As shown in Fig. 8(a), the center of GB A is the division point a instead of the center of the data points in A, which is computed using (1). However, as shown in step 5 in Fig. 7, the global division, that is, a division of all division points, is performed so that a division point is close to the center of the corresponding GB. For example, Fig. 9 shows the comparison results between the conventional GB generation method and our proposed acceleration method. Fig. 9(a) shows the results of the GB generation using the conventional method. Fig. 9(b) shows the experimental results obtained using the proposed acceleration method before the

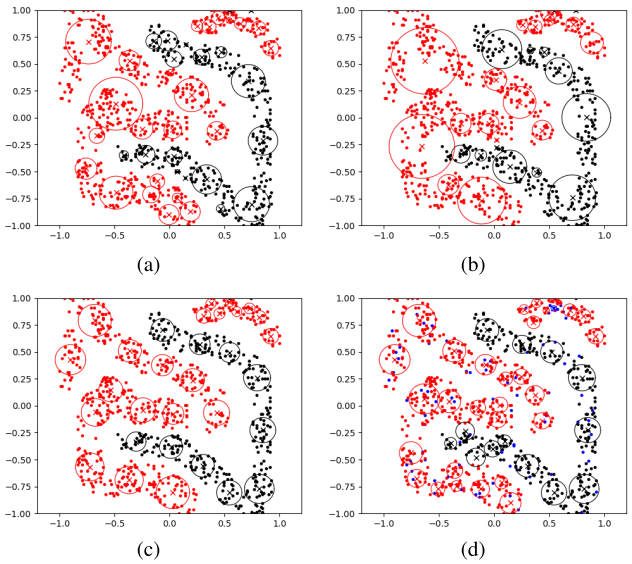


Fig. 9. Comparison of the distribution of GBs before and after global division. (a) Result of GB generation using k-means. (b) Result of the acceleration GB generation method without using the global division. (c) Distribution of GBs using the global division. (d) Experimental result using the proposed acceleration method in a noisy dataset where the noise points are colored with blue.

global division is performed. It can be seen from Fig. 9(c) that after the global division is performed, in comparison with those in Fig. 9(b), the division center of a GB, that is, its division point, changes to be closer to its true center computed using (1); thus, the points in the GB become more tightly and uniformly distributed, and the decision boundary is clearer. In addition, Fig. 9(d) shows the experimental results when label noise points were added. Fig. 9(d) shows that the GBs generated using the acceleration method can fit the division boundary well in the noisy data because of the robustness of GB computing. The design of the GB generating acceleration method is presented in Algorithm 1.

Algorithm 1 GB Generation Acceleration Method

Input: Dataset D, the purity threshold T
Output: The granular-balls

Treat the whole dataset D as a granular-ball A_1^i, where i is initialized to 1, and represents the number of iterations;
$balls$ were initialized to A_1^1;
Randomly select k-1 heterogeneous points as initial division centers on D, where k represents the number of classes in D, and compute the distances from all points to the division centers;
repeat
  for each $A_j^i \in balls$ **do**
    The purity T_j^i is equal to the percentage of majority samples in A_j^i;
    if $T_j^i < T$ **then**
      Randomly select k-1 heterogeneous points as the new division centers, where k represents the number of classes in A_j^i;
      Compute the distances from the points in the ball to the new division centers;
      Based on the distances in step 9, granular-balls $A_1^{i+1}, A_2^{i+1}, \ldots, A_k^{i+1}$ are generated;
      $balls = balls - \{A_j^i\}$
      $balls = balls + \{A_1^{i+1} + A_2^{i+1} + \cdots + A_k^{i+1}\}$;
    end if
  end for
until $|balls|$ does not increase.
Perform a global division.

C. Time Complexity

The time complexity of k-means is $O(Nkt)$ [39], where k represents the number of clusters, and t represents the number of iterations. The convergence speed of k-means is fast and can be considered approximately linear. The acceleration GB generation method only needs to compute the distance between the data in this cluster and the new division centers each time GBs are generated. Assuming a dataset with m classes of data, in the first round of splitting, the computation time is mN.

In the second round, $m(m-1)$ new division centers are generated, and the computation times are approximately

$$m(m-1) * \frac{N}{m} = (m-1)N. \tag{9}$$

In the third round, $m^2(m-1)$ division centers were newly generated, and the computation time was approximately

$$m^2(m-1) * \frac{N}{m^2} = (m-1)N. \tag{10}$$

In the fourth round, $m^3(m-1)$ division centers were newly generated, and the computation time was approximately

$$m^3(m-1) * \frac{N}{m^3} = (m-1)N. \tag{11}$$

Assuming a total of n iterations, the time complexity of the last global division is $O(kN)$, where k is the number of GBs and the total time complexity is $O((mn-n+k+1)N)$. However, it is worth noting that the GBs stopped splitting when the splitting conditions were not met. Most GBs stopped splitting halfway through. Therefore, the actual time complexity of the acceleration GB generation method is much lower than that of $O((mn-n+k+1)N)$. The time complexity of the acceleration method is linear, which prevents unnecessary calculations.

V. Adaptive GB Generation Method

A. Motivation

Using the incoming purity threshold parameter, the existing method can generate GBs that satisfy the purity threshold. The main problem with the existing method is that the purity threshold parameter cannot adapt to the data distribution of each dataset, and it is difficult to find a splitting standard that matches the data distribution for each dataset. To solve this problem, we propose a purity-adaptive GB generation

method based on the acceleration GB generation method. Purity adaptation is important for GB generation so that the GB generation process is completely parameter-free, and a completely parameter-free classifier, GBkNN, has been developed.

B. Adaptive Conditions of GB Splitting

In this section, we propose three adaptive conditions to realize the adaptive generation of GBs, including whether the weighted purity sum of child balls for each GB increases or not, whether there is overlap between any pair of heterogeneous GBs, and whether each GB reaches the lower bound of purity, that is, the purity of the initial GB of the whole dataset. The specific design is as follows.

1) Weighed Purity Sum of Child Balls: The purity was designed to measure the quality of the GB. Therefore, a direct idea is to design an indicator to measure the purity of the child balls. Then, whether a GB should be split is determined by whether its child GB's purity becomes larger than itself. Considering the fact that the more samples in the ball, the more important the ball is, we designed the weighed purity sum of the child balls to measure the purity of the child balls, as shown in Definition 3.

Definition 3: Given the GBs A and its child balls $A_i (i = 1, 2, \ldots, k)$, where $\bigcup_{i=1}^{k} A_i = A$ and $\bigcap_{i=1}^{k} A_i = \emptyset$. $|.|$ denotes the number of elements in a set. k denotes the number of classes in A. A_i^l denotes a set consisting of samples whose labels are equal to l, and A_i^*, that is, $l = *$, represents the set consisting of samples in the majority class in A_i. The weighted purity sum W of the child balls of A can be defined as follows:

$$\begin{aligned} W &= \frac{|A_1|}{|A|} \times \frac{|A_1^*|}{|A_1|} + \frac{|A_2|}{|A|} \times \frac{|A_2^*|}{|A_2|} + \cdots + \frac{|A_k|}{|A|} \times \frac{|A_k^*|}{|A_k|} \\ &= \sum_{i=1}^{k} \frac{|A_i^*|}{|A|} \\ &= \frac{\sum_{i=1}^{k} |A_i^*|}{|A|}. \end{aligned} \tag{12}$$

Based on Definition 3, Theorem 1 is proposed to describe the condition under which a GB should be split.

Theorem 1: Given a GB A, whose label is denoted by label(A) and purity by $T = (|A^*|/|A|)$, and its child GB $A_i (i = 1, 2, \ldots, k)$, where k is the number of child GBs. label$(A) = L$, W represents the weighted purity sum of A_i.

1) $\forall A_i \subset A$. If label(A) = label(A_i), then $W = T$.
2) $\exists A_i \subset A$; if label$(A) \neq$ label(A_i), then $W > T$.

Proof: 1) When $\forall A_i \subset A$ and label(A) = label(A_i), the majority samples in A are also the majority samples in all child balls; thus, we have

$$A^* = A^l \tag{13}$$

$$A_i^* = A_i^l. \tag{14}$$

At the same time, the majority of the samples in A are equal to the sum of the majority samples in all child balls. Combining with (13) and (14), we get

$$\sum_{i=1}^{k} A_i^* = \sum_{i=1}^{k} A_i^l = A^l = A^*. \tag{15}$$

From (15) and the Definition 3, we can easily get

$$W = \frac{\sum_{i=1}^{k} |A_i^*|}{|A|} = \frac{|A^*|}{|A|} = T. \tag{16}$$

So

$$W = T. \tag{17}$$

2) When $\exists A_i \subset A$ and label$(A) \neq$ label(A_i), the majority sample in A is not necessarily the majority sample of all child balls. Assuming $A_m (m = 1, 2, \ldots, k)$ has a different label from A, we obtain

$$|A_m^*| > |A_m^l|. \tag{18}$$

In addition, if the samples labeled l in the parent ball are equal to the sum of those in all the child balls, we have

$$|A^l| = \sum_{i=1}^{k} |A_i^l|. \tag{19}$$

Combining with (18) and (19), we get

$$\begin{aligned} \sum_{i=1}^{k} |A_i^*| &= |A_1^*| + |A_2^*| + \cdots + |A_m^*| + \cdots + |A_k^*| \\ &> |A_1^l| + |A_2^l| + \cdots + |A_m^l| + \cdots + |A_k^l| \\ &= \sum_{i=1}^{k} |A_i^l| \\ &= |A^l| \\ &= |A^*|. \end{aligned} \tag{20}$$

From (20) and the Definition 3, we get

$$W = \frac{\sum_{i=1}^{k} |A_i^*|}{|A|} > \frac{|A^*|}{|A|} = T. \tag{21}$$

So

$$W > T. \tag{22}$$

When W is greater than T, the label of some child balls is different from that of the parent ball, that is, the minority samples in the parent ball become the majority samples in some child balls. It can be concluded that the weighted purity sum of the child balls will be greater than that of the parent ball, and the number of correctly classified samples will increase. When W equals T, it represents a special case in which the parent ball and all the child balls have the same label.

As shown in Fig. 10, ball A in Fig. 10(a) is split into balls A_1 and A_2 in Fig. 10(b) using Theorem 1, and the labels of A and A_2 are different. At this time, the minority class in A becomes the majority of the classes in A_2, so the weighted purity sum of A_1 and A_2 will be greater than the purity of A. From the perspective of GBkNN, the number of samples with correct classification also increases. However, in this case, premature convergence will still occur because the accuracy does not increase monotonically. Fig. 10(b) shows a simple example of premature convergence using only the conditions in Section V-B1, that is, there is an overlap between heterogeneous GBs.

To this end, we introduce the second condition that there can be no overlap between the heterogeneous GBs.

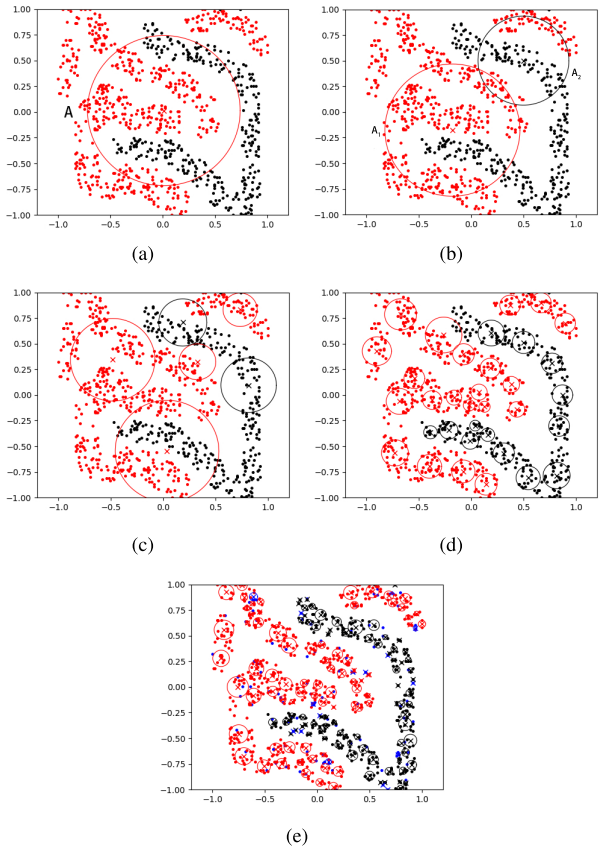


Fig. 10. Situation before and after splitting using the adaptive GB generation method. (a) Parent ball before splitting. (b) Child balls after splitting. (c) Situation after de-overlap. (d) Algorithm convergence result using the proposed adaptive method. (e) Experimental result using the proposed adaptive method in a noisy dataset where the noise points are colored with blue.

2) De-Overlap Between Heterogeneous GBs: For the problem of GB overlap, it is necessary to further detect whether there are heterogeneous GBs based on Section V-B1, and further split and refine the overlapping GBs to make the decision boundary clearer. To improve efficiency, the next round of overlap detection only needs to traverse the child GBs of the overlapping GBs. The boundary overlap problem of heterogeneous GBs is defined as follows:

$$\begin{aligned} &\exists \|c_i - c_j\| \le \|r_i + r_j\|, \text{for } i, j \in \{1, 2, \ldots, m\} \\ &\text{s.t. } \text{label}(\text{GB}_i) \ne \text{label}\big(\text{GB}_j\big) \end{aligned} \tag{23}$$

where c_i represents the center of the ith GB, r_i represents the radius of the ith GB, and m is the total number of GBs. In addition, the effect of the condition "$\text{label}(\text{GB}_i) \ne \text{label}(\text{GB}_j)$" concentrates the boundary overlap problem between heterogeneous GBs and reduces the cost of computing and analyzing the problem. The overlap between GBs of the same type does not affect the decision boundary. As shown in Fig. 10(c), it can be seen that, compared to Fig. 10(b), the GBs after de-overlap are more suitable for the data distribution.

3) Adaptive Purity Lower Bound: Additionally, the purity of the GBs should have an adaptive lower bound. The lower bound is the proportion of the initial majority of the samples of the total sample, that is, the purity of the initial GB. As shown in Fig. 10(c), the GBs adaptively generated GBs with a purity lower than the initial purity. The quality of such GBs is extremely low, which reduces the classification accuracy of the final GBkNN. For minority samples, the proportion of incorrectly classified samples can be considered the noise rate. That is, the purity of all the GBs must be greater than the purity of the initial GB.

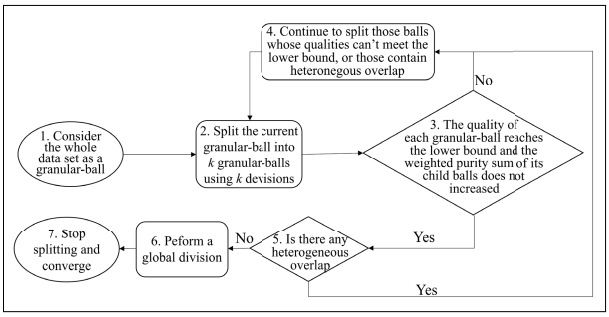


Fig. 11. Basic idea of the adaptive GB generation method.

Fig. 10(d) shows the result of the GB generation using the adaptive method. The two-colored GBs in the figure represent the two classes of data. In addition, Fig. 10(e) shows the result of the GB generation using the adaptive method when label noise points are added. The blue points in the figure represent noisy data, and the other two-colored points represent the two original data points. Noise points were generated by randomly changing the labels of the samples in the dataset. The optimization goal of the adaptive GB generation method can be expressed as

$$\begin{aligned} &\text{Min } \lambda_1 * n / \sum_{j=1}^{m} \big(|\text{GB}_j|\big) + \lambda_2 * m \\ &\text{s.t. } \text{quality}\big(\text{GB}_j\big) > T_0, W\big(\text{GB}'_j\big) > \text{quality}\big(\text{GB}_j\big) \\ &\quad \|c_i - c_j\| > \|r_i + r_j\| \\ &\quad \times \big(i, j \in [1, m] \text{label}(\text{GB}_i) \ne \text{label}\big(\text{GB}_j\big)\big) \end{aligned} \tag{24}$$

where λ_1 and λ_2 are the corresponding weight coefficients and c_i, r_i represent the center and radius of GB_i, respectively. T_0 denotes the adaptive purity lower bound of the GBs, and GB'_j represents the child GBs of GB_j. In addition, $\text{label}(\text{GB}_i)$ and W are mentioned above.

C. Method Design

The basic concept of the GB generation of the adaptive GB generation method is shown in Fig. 11.

In step 2 of Fig. 11, based on the accelerated GB generation method, k division is used to split the GB, where k denotes the number of classes in a GB. Therefore, the computational cost is directly decreased. In addition, as shown in step 3, when the weighted purity sum of the child balls is greater than the purity of its parent ball and the purity of the GB reaches the lower bound, the child balls are retained and there is overlap between heterogeneous GBs. As shown in Fig. 10(d), the boundary of the GBs when the algorithm converged was consistent with that of the dataset.

The algorithm design for the adaptive GB generation method is presented in Algorithm 2.

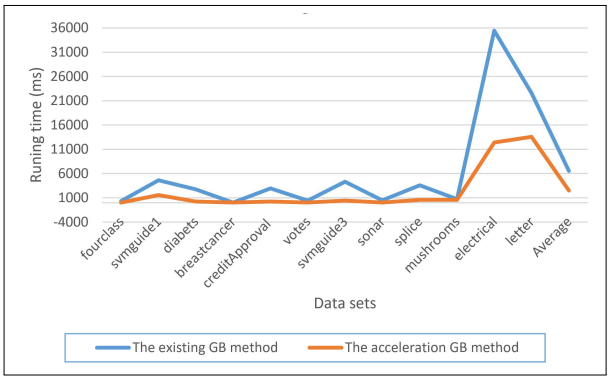


Fig. 12. Comparison of the running time of the acceleration GB generation method and the existing method.

Algorithm 2 GB Generation Adaptive Method

Input: Dataset D

Output: The granular-balls

Treat the whole dataset D as a granular-ball A_1^i, where i is initialized to 1, and represents the number of iterations;
$balls$ were initialized to A_1^1;
repeat
  for each $A_j^i \in balls$ **do**
    Implement the acceleration granular-ball generation method on A_j^i, and pre-generate k granular-balls $A_1^{i+1}, A_2^{i+1}, \ldots, A_k^{i+1}$, where k represents the number of classes in the ball;
    The T_j^i represents the purity of A_j^i;
    The W_j^i represents the weighted purity sum of the child balls of A_j^i;
    if $W_j^i > T_j^i$ or $T_j^i <= T_1^1$ **then**
      $balls = balls - \{A_j^i\}$
      $balls = balls + \{A_1^{i+1} + A_2^{i+1} + \cdots + A_k^{i+1}\}$;
    end if
  end for
  De-overlap between heterogeneous granular-balls;
until $|balls|$ does not increase
Perform a global division.

D. Discussion

As we know, the original GB generation method needs to introduce a parameter of purity threshold; therefore, this method is not adaptive. At the same time, the original GBkNN algorithm is based on the original GB generation method; therefore, the original GBkNN is also not adaptive. After the proposed adaptive GB generation method in this article, the GBkNN algorithm can be improved based on this adaptive method to make it a fully adaptive kNN algorithm.

VI. Experiments

To demonstrate the feasibility and effectiveness of the acceleration GB generation method and the adaptive GB generation method, we compared them with kNN and two popular or state-of-the-art methods based on granular computing, GBkNN [17] and GBS [38]. Because of the robustness of the GB, our experiments were carried out on both the raw and noisy datasets. We verified the performance of the acceleration GB generation method, adaptive GB generation method, and efficiency of the acceleration method. We randomly selected ten real datasets from the University of California, Irvine (UCI) benchmark datasets, as shown in the following tables. Experimental hardware environment: PC with an Intel Core i7-107000 CPU@2.90 GHz with 32 GB RAM. Experimental software environment: Python 3.9. Table I lists the datasets.

TABLE I
Information About the Datasets

Data sets	Number of the samples	dimensionality
fourclass	862	2
svmguide1	7089	4
diabetes	768	8
breastcancer	683	10
creditApproval	690	15
votes	435	16
svmguide3	1284	22
sonar	208	60
splice	1000	60
mushrooms	8124	112
electrical	10000	13
letter	20000	16

TABLE II
Comparison of Average Test Accuracy (Raw Datasets)

Data sets	Acc$^+$		Adp		Origin		kNN
	mean	max	mean	max	mean	max	
fourclass	0.990	0.987	0.988	0.973	0.990	0.958	**0.999**
svmguide1	0.959	0.965	0.930	**0.970**	0.960	0.796	0.960
diabetes	0.824	**0.838**	0.834	0.824	0.718	0.697	0.748
breastcancer	0.950	0.972	0.977	0.965	**0.982**	0.962	0.973
creditApproval	**0.770**	0.769	0.722	0.714	0.669	0.599	0.659
votes	0.906	0.917	0.868	**0.922**	0.871	0.722	0.875
svmguide3	**0.831**	0.828	0.812	0.818	0.786	0.776	0.788
sonar	**0.895**	0.886	0.833	0.855	0.833	0.757	0.831
splice	0.745	**0.807**	0.765	0.796	0.605	0.595	0.681
mushrooms	0.994	**1.000**	**1.000**	**1.000**	0.993	0.715	**1.000**
electrical	0.824	0.830	0.792	0.790	**0.878**	0.731	0.780
letter	0.968	0.967	0.887	**0.986**	0.900	0.739	0.976
Average	0.888	**0.897**	0.870	0.883	0.849	0.754	0.856

TABLE III
Comparison of Average Test Accuracy After Sampling With GBS (Raw Datasets)

Data sets	Origin-GBS	Acc$^+$-GBS	Adp-GBS	Adp-Base	kNN
fourclass	0.9890	0.9902	0.9942	0.9936	**0.9971**
svmguide1	0.9558	0.9612	0.9587	**0.9643**	0.9596
diabetes	0.7331	**0.7494**	0.7448	0.7260	0.7312
breastcancer	0.9644	0.9585	**0.9696**	0.9637	0.9644
creditApproval	**0.6855**	0.6725	0.6609	0.6841	0.6623
votes	0.8884	0.9000	0.9029	**0.9072**	0.8870
svmguide3	0.7835	0.7803	**0.7863**	0.7787	0.7807
sonar	0.8048	**0.8476**	0.8262	0.8286	0.8048
splice	0.6964	**0.7265**	0.7061	0.7245	0.6750
mushrooms	0.9994	**1.0000**	**1.0000**	**1.0000**	**1.0000**
electrical	0.7817	0.7785	**0.7819**	0.7725	0.7775
letter	0.9732	0.9751	**0.9781**	0.9731	0.9774
Average	0.8546	**0.8617**	0.8591	0.8597	0.8514

TABLE IV

COMPARISON OF THE NUMBER OF GBS GENERATED BY THE ACCELERATION GB GENERATION METHOD AND THE EXISTING METHOD

Data sets	fourclass	svmguide1	diabetes	breastcancer	creditApproval	votes	svmguide3	sonar	splice	mushrooms	electrical	letter	Average
balls	**31**	533	394	**2**	426	61	597	**69**	517	**14**	4637	**2512**	816
balls+	**31**	**360**	**348**	25	**364**	**60**	**513**	72	**500**	44	**4056**	2960	**778**

A. Experiments on Raw Datasets

In this section, we split the dataset into ten parts, take one part for testing, and use the test accuracy as the evaluation index to verify the effectiveness of the acceleration and adaptive GB generation methods. Because GB generation still has certain randomness, we performed experiments on each method ten times and took the average classification accuracy of the ten experiments results for comparison. The kNN method uses a tenfold cross-validation result.

Table II lists the experimental accuracy of kNN under noise-free conditions. "Acc$^+$," "Adp," "Origin," and kNN represent the acceleration GB generation method, the adaptive GB generation method, the existing GB generation method, and kNN, respectively. "mean" and "max" represent the experimental accuracy of the average distance and the maximum distance as the radius of the GB. The two proposed methods, the acceleration GB generation method and the adaptive GB generation method, obtained better performance in terms of accuracy compared to the existing method and kNN. The decision boundary obtained using existing methods is still not sufficiently clear. Therefore, when measuring kNN accuracy, the GB closest to the test point is likely to be inaccurate. The global division was performed after the splitting of the GB was stopped. Therefore, the two methods proposed in this study can obtain a higher kNN accuracy than the existing methods in the raw dataset.

The column 1–3 in Table III is based on the GBS method. Column 4 shows the GBS accuracy when applying the adaptive strategy to the original method, that is, the original k-means method is still used in the division process of the GBs rather than the acceleration method. First, we use the first four methods in Table III to generate the GBs, and then the GBS method is used to sample the generated GBs. Finally, kNN is used to classify the sampled result. The last column represents the direct classification of the raw dataset using kNN. According to [38], the purity was set from 0.54 to 1.0, with a step size of 0.2 GBS algorithm. It can be observed that our methods have a higher accuracy on most datasets than the other methods.

To demonstrate the efficiency of the acceleration GB generation method, we chose the existing GB generation method for comparison. Fig. 12 shows the running time of the two methods on raw datasets, and the time unit is ms. Table IV shows the comparison of the number of GBs generated by the acceleration method and the existing method on raw datasets, where "ball+" and "ball" denote the acceleration method and the existing method, respectively. Compared with the existing GB generation method from Fig. 12 and Tables II–IV, the acceleration method has similar accuracy and higher efficiency on most datasets while generating a similar number of GBs.

TABLE V

COMPARISON OF AVERAGE TEST ACCURACY AFTER SAMPLING WITH GBS (NOISE RATE 10%)

Data sets	Origin-GBS	Acc$^+$-GBS	Adp-GBS	Adp-Base	kNN
fourclass	**0.8815**	0.8792	0.8763	0.8717	0.8769
svmguide1	0.8523	**0.8625**	0.8461	0.8468	0.8428
diabetes	0.6721	**0.6935**	0.6701	0.6662	0.6578
breastcancer	**0.8711**	0.8504	0.8593	0.8511	0.8393
creditApproval	**0.6442**	0.6225	0.6283	0.6213	0.6123
votes	**0.8188**	0.8029	0.7957	0.8116	0.7957
svmguide3	**0.7297**	0.7285	0.7088	0.6976	0.6964
sonar	**0.7571**	0.7357	0.7452	0.7471	**0.7571**
splice	0.6449	**0.6485**	0.6388	0.6291	0.6173
mushrooms	**0.8926**	0.8781	0.8358	0.8445	0.8740
electrical	**0.7218**	0.6932	0.7024	0.6992	0.6946
letter	0.8567	**0.8574**	0.8544	0.8554	0.8534
Average	**0.7786**	0.7710	0.7634	0.7618	0.7598

TABLE VI

COMPARISON OF AVERAGE TEST ACCURACY AFTER SAMPLING WITH GBS (NOISE RATE 20%)

Data sets	Origin-GBS	Acc$^+$-GBS	Adp-GBS	Adp-Base	kNN
fourclass	0.7370	**0.7948**	0.7583	0.7792	0.7046
svmguide1	0.7602	**0.7677**	0.7098	0.7331	0.7156
diabetes	0.6214	**0.6390**	0.6065	0.5740	0.5994
breastcancer	0.7652	**0.7941**	0.7815	0.7267	0.6852
creditApproval	**0.6210**	0.5775	0.5695	0.5667	0.5696
votes	0.7217	0.7072	0.7014	**0.7232**	0.6725
svmguide3	0.6663	**0.6775**	0.6261	0.6281	0.6108
sonar	**0.6786**	0.6238	0.6452	0.6333	0.6786
splice	**0.5929**	0.5857	0.5740	0.5709	0.5699
mushrooms	**0.7830**	0.7584	0.6985	0.7570	0.7314
electrical	**0.6590**	0.6151	0.6205	0.6155	0.6148
letter	**0.7444**	0.7408	0.7185	0.7171	0.7228
Average	**0.6959**	0.6935	0.6675	0.6687	0.6563

B. Experiments on Noisy Datasets

Each dataset has four class noise rates: 10%, 20%, 30%, and 40%. Noise is generated by changing the labels of randomly selected samples in a dataset. Tables V–VIII show the GBS's highest average test accuracy obtained from the purity optimization of the existing method under different noise rates, and the GBS's highest average accuracy of the adaptive and acceleration GB generation methods. The purity of the GBS algorithm was also set from 0.54 to 1.0, with a step size of 0.2 in the GBS algorithm. The acceleration and adaptive methods adopt the strategy of selecting heterogeneous sample points as new clustering centers when splitting a GB. This can make the algorithm converge faster, but will reduce the accuracy when dealing with noisy datasets. It can also be seen from the experimental results of noisy datasets that the acceleration GB generation method and the adaptive GB generation method can obtain a similar law to the existing GB generation method on noisy data, that is, when the noise rate in the dataset is larger, the advantage of the original kNN is

TABLE VII
COMPARISON OF AVERAGE TEST ACCURACY AFTER SAMPLING WITH GBS (NOISE RATE 30%)

Data sets	Origin-GBS	Acc$^+$-GBS	Adp-GBS	Adp-Base	kNN
fourclass	**0.6711**	0.6659	0.6653	0.6497	0.6156
svmguide1	0.6543	**0.6809**	0.6025	0.6250	0.6019
diabetes	0.5669	**0.5877**	0.5370	0.5357	0.5266
breastcancer	**0.7052**	0.6800	0.6467	0.5896	0.6178
creditApproval	0.5667	**0.5688**	0.5355	0.5188	0.5355
votes	**0.6435**	0.6362	0.6232	0.6145	0.5928
svmguide3	**0.6116**	0.6096	0.5735	0.5502	0.5434
sonar	0.5690	**0.5786**	0.5667	0.5524	0.5690
splice	**0.5592**	0.5500	0.5342	0.5214	0.5245
mushrooms	**0.6881**	0.6495	0.5921	0.6314	0.6151
electrical	**0.6019**	0.5971	0.5573	0.5593	0.5538
letter	**0.6432**	0.6386	0.6058	0.6154	0.6073
Average	**0.6234**	0.6202	0.5867	0.5803	0.5753

TABLE VIII
COMPARISON OF AVERAGE TEST ACCURACY AFTER SAMPLING WITH GBS (NOISE RATE 40%)

Data sets	Origin-GBS	Acc$^+$-GBS	Adp-GBS	Adp-Base	kNN
fourclass	**0.5775**	0.5468	0.5740	0.5480	0.5312
svmguide1	0.5711	**0.5807**	0.5340	0.5395	0.5316
diabetes	0.5117	**0.5468**	0.5026	0.5000	0.4890
breastcancer	**0.5904**	0.5778	0.5615	0.5504	0.5230
creditApproval	0.5080	**0.5435**	0.5145	0.5065	0.4819
votes	**0.5841**	0.5652	0.5362	0.5652	0.5319
svmguide3	**0.5627**	0.5498	0.5171	0.5313	0.5104
sonar	0.4881	0.5405	**0.5690**	0.5238	0.4833
splice	**0.5321**	0.5179	0.5184	0.5026	0.5281
mushrooms	**0.5860**	0.5641	0.5255	0.5339	0.5338
electrical	**0.5461**	0.5363	0.5175	0.5259	0.5138
letter	**0.5606**	0.5526	0.5285	0.5376	0.5237
Average	**0.5515**	0.5518	0.5332	0.5304	0.5151

more obvious. The two groups of experiments of the adaptive method have approximate accuracy. It can be concluded that the acceleration method has similar accuracy to the original method when significantly improving the efficiency. However, the adaptive method still shows slightly lower accuracy than the existing method when dealing with noisy datasets. The adaptive method significantly improves upon the existing method to make it adaptive.

C. Discussion

From the above experiments, it can be concluded that the adaptive GB generation method still has certain defects, and it still cannot adaptively generate optimal GBs. The reason why the accuracy of the adaptive method is slightly lower than that of the original method is as follows: the lower bound of the adaptive purity of GBs in the adaptive method may be too low, resulting in some GBs of poor quality that affects the overall accuracy, and some poor-quality GBs fail to generate child balls whose weighted purity sum is greater than that of the parent ball and premature convergence. Because of the above problems, this study provides only an enlightening strategy for the adaptive generation of GBs.

VII. CONCLUSION AND FUTURE WORK

This article proposes a method for accelerating GB generation, which can greatly improve the efficiency of GB generation while ensuring accuracy. Simultaneously, a new GB clustering method, that is, the adaptive GB generation method, is proposed. This adaptive method avoids the problem that the existing method needs to manually set the purity threshold parameter and makes the generation process of GBs completely adaptive. Experiments show that the acceleration method performs better than the adaptive method for both noisy and non-noisy data.

At the same time, as shown by the experiments, the experimental accuracy of the adaptive method was slightly lower than that of the existing method. This proves that our method is effective, but whether there are other adaptive methods, such as those based on the consistency of the internal distribution of GBs, may develop a more effective GB adaptive optimization method. However, the proposed methods exhibit lower accuracy in some cases than the existing method; therefore, we will study how to improve their accuracy in future work.

REFERENCES

[1] L. Chen, "Topological structure in visual perception," *Science*, vol. 218, no. 4573, pp. 699–700, Nov.1982.
[2] L. A. Zadeh, "Fuzzy sets and information granularity," *Adv. Fuzzy Set Theory Appl.*, vol. 11, pp. 3–18, Sep. 1979.
[3] L. A. Zadeh, "Toward a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic," *Fuzzy Sets Syst.*, vol. 90, no. 2, pp. 111–127, 1997.
[4] E. Backer and A. K. Jain, "A clustering performance measure based on fuzzy set decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-3, no. 1, pp. 66–75, Jan. 1981.
[5] B. Kosko, "Counting with fuzzy sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 4, pp. 556–557, Jul. 1986.
[6] W.-B. Zhang and G.-Y. Zhu, "A multiobjective optimization of PCB prototyping assembly with OFA based on the similarity of intuitionistic fuzzy sets," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 7, pp. 2054–2061, Jul. 2020.
[7] X. Liu and S. Wan, "Combinatorial iterative algorithms for computing the centroid of an interval type-2 fuzzy set," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 4, pp. 607–617, Apr. 2019.
[8] S. Xia, H. Zhang, W. Li, G. Wang, E. Giem, and Z. Chen, "GBNRS: A novel rough set algorithm for fast adaptive attribute reduction in classification," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 3, pp. 1231–1242, Mar. 2020, doi: 10.1109/TKDE.2020.2997039.
[9] J. Liang, F. Wang, C. Dang, and Y. Qian, "A group incremental approach to feature selection applying rough set technique," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 294–308, Feb. 2012.
[10] Y. Qian, J. Liang, W. Pedrycz, and C. Dang, "Positive approximation: An accelerator for attribute reduction in rough set theory," *Artif. Intell.*, vol. 174, nos. 9–10, pp. 597–618, 2010.
[11] Q. Hu, L. Zhang, Y. Zhou, and W. Pedrycz, "Large-scale multimodality attribute reduction with multi-kernel fuzzy rough sets," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 1, pp. 226–238, Feb. 2017.
[12] L. Zhang and B. Zhang, "The quotient space theory of problem solving," *Fundamenta Informaticae*, vol. 59, nos. 2–3, pp. 287–298, 2004.
[13] B. Zhang and L. Zhang, "Theory of fuzzy quotient space (methods of fuzzy granular computing)," *J. Softw.*, vol. 14, no. 4, pp. 770–776, 2003.
[14] D. Li, D. Cheung, X. Shi, and V. Ng, "Uncertainty reasoning based on cloud models in controllers," *Comput. Math. Appl.*, vol. 35, no. 3, pp. 99–123, Feb. 1998.
[15] D. Li, C. Liu, and W. Gan, "A new cognitive model: Cloud model," *Int. J. Intell. Syst.*, vol. 24, no. 3, pp. 357–375, Mar. 2009.
[16] S. Xia *et al.*, "A fast adaptive K-means with no bounds," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 87–99, Jan. 2020, doi: 10.1109/TPAMI.2020.3008694.
[17] S. Xia, Y. Liu, X. Ding, G. Wang, H. Yu, and Y. Luo, "Granular ball computing classifiers for efficient, scalable and robust learning," *Inf. Sci.*, vol. 483, pp. 136–152, May 2019.
[18] G. Y. Wang, "Rough reduction in algebra view and information view," *Int. J. Intell. Syst.*, vol. 18, no. 6, pp. 679–688, Jun. 2003.
[19] D. Pei and D. Miao, "From soft sets to information systems," in *Proc. IEEE Int. Conf. Granular Comput.*, vol. 2, Aug. 2005, pp. 617–621.

[20] Z. Xu and Q. Wang, "On the properties of covering rough sets model," *J. Henan Normal Univ. Natural Sci.*, vol. 33, no. 1, pp. 130–132, 2005.
[21] Y. Yao, "Decision-theoretic rough set models," in *Proc. Int. Conf. Rough Sets Knowl. Technol.* Toronto, ON, Canada: Springer, 2007, pp. 1–12.
[22] Q. H. Hu, L. Zhang, and S. An, "On robust fuzzy rough set models," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 4, pp. 636–651, Dec. 2011.
[23] S. An, Q. Hu, W. Pedrycz, P. Zhu, and E. C. Tsang, "Data-distribution-aware fuzzy rough set model and its application to robust classification," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 3073–3085, Nov. 2015.
[24] W.-Z. Wu and Y. Leung, "Theory and applications of granular labelled partitions in multi-scale decision tables," *Inf. Sci.*, vol. 181, no. 18, pp. 3878–3897, Sep. 2011.
[25] H. Chen, T. Li, D. Ruan, J. Lin, and C. Hu, "A rough-set-based incremental approach for updating approximations under dynamic maintenance environments," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 2, pp. 274–284, Feb. 2011.
[26] S. Salehi, A. Selamat, M. R. Mashinchi, and H. Fujita, "The synergistic combination of particle swarm optimization and fuzzy sets to design granular classifier," *Knowl.-Based Syst.*, vol. 76, pp. 200–218, Mar. 2015.
[27] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 7–19, Jan. 1967.
[28] W.-Y. Loh, "Classification and regression trees," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 14–23, Jan. 2011.
[29] S. Dick and A. Kandel, "Granular computing in neural networks," in *Granular Computing*. Heidelberg, Germany: Springer, 2001, pp. 275–305.
[30] D. Leite, P. Costa, and F. Gomide, "Evolving granular neural networks from fuzzy data streams," *Neural Netw.*, vol. 38, pp. 1–16, Feb. 2013.
[31] M. M. Gupta and G. Knopf, "Fuzzy neural network approach to control systems," in *Proc. Int. Symp. Uncertainty Modeling Anal.*, Dec. 1990, pp. 483–488.
[32] F.-Y. Wang and H.-M. Kim, "Implementing adaptive fuzzy logic controllers with neural networks: A design paradigm," *J. Intell. Fuzzy Syst.*, vol. 3, no. 2, pp. 165–180, 1995.
[33] M. Syeda, Y.-Q. Zhang, and Y. Pan, "Parallel granular neural networks for fast credit card fraud detection," in *Proc. IEEE World Congr. Comput. Intell. IEEE Int. Conf. Fuzzy Syst.*, vol. 1, May 2002, pp. 572–577.
[34] H. S. Park, W. Pedrycz, and S. K. Oh, "Granular neural networks and their development through context-based clustering and adjustable dimensionality of receptive fields," *IEEE Trans. Neural Netw.*, vol. 20, no. 10, pp. 1604–1616, Oct. 2009.
[35] Y. Tang and Y. He, "Granular support vector machine with random granularity," U.S. Patent 8 160 975, Apr. 17, 2012.
[36] W. Pedrycz and G. Vukovich, "Granular neural networks," *Neurocomputing*, vol. 36, pp. 205–224, Feb. 2001.
[37] G. Wang, "DGCC: Data-driven granular cognitive computing," *Granular Comput.*, vol. 2, no. 4, pp. 343–355, 2017.
[38] S. Xia, S. Zheng, G. Wang, X. Gao, and B. Wang, "Granular ball sampling for noisy label classification or imbalanced classification," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Aug. 30, 2021, doi: 10.1109/TNNLS.2021.3105984.
[39] Y. Zhou, H. Yu, and X. Cai, "A novel K-Means algorithm for clustering and outlier detection," in *Proc. 2nd Int. Conf. Future Inf. Technol. Manage. Eng.*, Dec. 2009, pp. 476–480.

Shuyin Xia (Member, IEEE) received the B.S. and M.S. degrees in computer science from the Chongqing University of Technology, Chongqing, China, in 2008 and 2012, respectively, and the Ph.D. degree from the College of Computer Science, Chongqing University, Chongqing, in 2015.

He is currently a Professor with the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications (CQUPT), Chongqing. He is also the Executive Deputy Director of the Big Data and Network Security Joint Laboratory, CQUPT. He has published more than 30 papers in journals and conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (T-PAMI), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (T-KDE), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (T-NNLS), IEEE TRANSACTIONS ON CYBERNETICS (T-CYB), and *Information Sciences*. His research interests include classifiers and granular computing.

Xiaochuan Dai received the B.S. degree from the Sichuan University of Arts and Science, Dazhou, China, in 2020, majoring in digital media technology. He is currently pursuing the M.S. degree in computer technology with the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China.

His research interests include granular computing and data mining.

Guoyin Wang (Senior Member, IEEE) received the B.E. degree in computer software, the M.S. degree in computer software, and the Ph.D. degree in computer organization and architecture from Xi'an Jiaotong University, Xi'an, China, in 1992, 1994, and 1996, respectively.

He worked with the University of North Texas, Denton, TX, USA; and the University of Regina, Regina, SK, Canada, as a Visiting Scholar from 1998 to 1999. Since 1996, he has been with the Chongqing University of Posts and Telecommunications, Chongqing, China, where he is currently a Professor and a Ph.D. Supervisor, the Director of the Chongqing Key Laboratory of Computational Intelligence, and the Dean of the Graduate School. He has published more than 300 papers in journals and conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (T-PAMI), IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (T-KDE), IEEE TRANSACTIONS ON IMAGE PROCESSING (T-IP), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (T-NNLS), and IEEE TRANSACTIONS ON CYBERNETICS (T-CYB). His research interests include data mining, machine learning, rough sets, granular computing, and cognitive computing.

Dr. Wang is the Steering Committee Chair of the International Rough Set Society (IRSS), the Vice-President of the Chinese Association for Artificial Intelligence (CAAI), and a Council Member of the China Computer Federation (CCF).

Xinbo Gao (Senior Member, IEEE) received the B.Eng., M.Sc., and Ph.D. degrees in signal and information processing from Xidian University, Xi'an, China, in 1994, 1997, and 1999, respectively.

From 1997 to 1998, he was a Research Fellow with the Department of Computer Science, Shizuoka University, Shizuoka, Japan. From 2000 to 2001, he was a Post-Doctoral Research Fellow with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong. From 2001 to 2020, he has been with the School of Electronic Engineering, Xidian University. He is currently the President of the Chongqing University of Posts and Telecommunications, Chongqing, China. He has published six books and approximately 200 technical papers in prestigious journals and conferences, including IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE (T-PAMI), IEEE TRANSACTIONS ON IMAGE PROCESSING (T-IP), IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (T-NNLS), IEEE TRANSACTIONS ON MEDICAL IMAGING (T-MI), Conference and Workshop on Neural Information Processing Systems (NIPS), IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE International Conference on Computer Vision (ICCV), The Association for Advancement of Artificial Intelligence (AAAI), and International Joint Conference on Artificial Intelligence (IJCAI).

Elisabeth Giem received the bachelor's degree in pure mathematics and the bachelor's degree in music (concentration in performance) from the University of California at Riverside (UCR), Riverside, CA, USA, the master's degree in computational and applied mathematics from Rice University, Houston, TX, USA, in 2005, and the master's degree in pure mathematics from UCR, in 2010, where she is currently pursuing the Ph.D. degree computer science with the Zizhong Chen's Supercomputing Laboratory (SuperLab).

Her research interests include, but are not limited to, high-performance computing, parallel and distributed systems, big data analytics, computational entomology, and numerical linear algebra algorithms and software.

Dr. Giem has been awarded the NICE:NRT in Integrated Computational Entomology Fellowship.