GBRS: A Unified Granular-Ball Learning Model of Pawlak Rough Set and Neighborhood Rough Set

Shuyin Xia, Member, IEEE, Cheng Wang, Guoyin Wang, Senior Member, IEEE, Xinbo Gao, Senior Member, IEEE, Weiping Ding, Senior Member, IEEE, Jianhang Yu, Yujia Zhai, Member, IEEE, and Zizhong Chen, Senior Member, IEEE

Abstract—Pawlak rough set (PRS) and neighborhood rough set (NRS) are the two most common rough set theoretical models. Although the PRS can use equivalence classes to represent knowledge, it is unable to process continuous data. On the other hand, NRSs, which can process continuous data, rather lose the ability of using equivalence classes to represent knowledge. To remedy this deficit, this article presents a granular-ball rough set (GBRS) based on the granular-ball computing combining the robustness and the adaptability of the granular-ball computing. The GBRS can simultaneously represent both the PRS and the NRS, enabling it not only to be able to deal with continuous data and to use equivalence classes for knowledge representation as well. In addition, we propose an implementation algorithm of the GBRS by introducing the positive region of GBRS into the PRS framework. The experimental results on benchmark datasets demonstrate that the learning accuracy of the GBRS has been significantly improved compared with the PRS and the traditional NRS. The GBRS also outperforms nine popular or the state-of-the-art feature selection methods. We have open-sourced all the source codes of this article at http://www.ee.cqupt.edu.cn/GBRS.html, https://github.com/syxiaa/GBRS.

Index Terms—Attribute reduction, feature selection, granular ball, granular computing, neighborhood rough set (NRS).

I. INTRODUCTION

With the advent of the big data era, both the feasible size and the dimensionality of data have increased dramatically [1], [2], [3], [4]. In machine learning and its related fields, the high dimensionality slows down the training speeds of models and increases the difficulty of the learning task. A higher dimensionality can lead to overfitting, which reduces the generalizability of a model [5], [6], [7], [8]. At high dimensions, the Euclidean distance fails as a usable metric, limiting the application range of models that rely on the Euclidean distance [9], [10], [11], [12], [13], [14].

Feature engineering was developed in response to these substantial problems [15], [16]. Feature selection, one of the most well-known methods in feature engineering, aims to select a subset of the feature set that can replace the original feature set. Feature selection possesses significant remedial effects: it can reduce feature dimension, increasing the training speed of a model; it can prevent overfitting, improving the generalizability of a model; and it can also increase the correlation between features and predictions, making a model more interpretive [17], [18], [19], [20].

Feature selection methods can be grouped into three categories: filter methods, wrapper methods, and embedded methods. Filter methods score features according to evaluation criteria. They then sort the features in a descending order according to the assigned score. Evaluation criteria usually fall in one of four categories: distance measures, information measures, dependence measures, and consistency measures [21]. Wrapper methods treat feature selection as a feature subset optimization process and use a classifier to evaluate feature subsets. Since each feature subset needs to train a classifier, most wrapper methods are inefficient. Accordingly, prior research on wrapper methods typically focuses on its optimization process. Embedded methods embed feature selection into the training process of the learning algorithm to screen out those features that are important to the model training [22], [23].

Feature selection algorithms based on the rough set theory rely on attribute reduction [24], [25], [26], and therefore, they are classified as the filter method. The rough set theory was proposed by Polish scientist Pawlak [27]. It is an effective mathematical tool to process uncertain, inconsistent, and incomplete data and has been widely applied in data
mining, machine learning, decision support systems, and other application fields [28], [29], [30], [31], [32], [33], [34]. The classical rough set is also called Pawlak rough set (PRS). The PRS theory divides samples into several equivalence classes and then defines upper and lower approximation sets using the union of the equivalence classes [27]. The upper and lower approximations are used to describe and approximate uncertain concepts, while samples are divided into positive region, boundary region, and negative region during this process. The number of samples in the positive region is used to measure the dependence of a label on the feature set, which is the score of the feature set. Heuristic attribute reduction algorithms based on the rough set theory can effectively reduce the time complexity of high-dimensional problems, making it an increasingly popular research topic in recent decades [35], [36], [37], [38], [39], [40], [41].

PRS and neighborhood rough set (NRS) are two most popular rough set theories. In the feature selection process, PRS granulates a dataset based on equivalence classes. An equivalence class consists of a set of attributes and a set of objects and can describe certain knowledge. Therefore, it provides a good interpretability. However, PRS is only able to process discrete data, while the real-world data are mostly continuous. Since the discretization will inevitably lead to information loss, there presents a serious hindrance to the development and application of rough set theory. Lin and Yao [42] conducted a series of studies on rough set and neighborhood system and analyzed the relation between neighborhood, rough set, and fuzzy set; Yao [43] used both the rough set and neighborhood systems to study the granular structure. Then, Hu et al. [45] developed Lin’s neighborhood model [44] and proposed the NRS model. NRS uses a neighborhood relation instead of an equivalence relation to granulate datasets, which enables NRS to process continuous data directly. However, in the model of NRS, the upper and lower approximations of the NRS consist of sample points instead of equivalence classes, which hinders the interpretability of NRS. In addition, the inconsistency between PRS and NRS further makes the rough set theory not sufficiently concise.

To elaborate, our main contributions are given as follows.

1) We propose a novel rough set model—granular-ball rough set (GBRS), which unifies PRS and NRS by introducing granular ball in the rough set theory. Consequently, GBRS is the first rough set model that can naturally process continuous data while maintaining the interpretability of equivalence classes.

2) Based on GBRS, we further propose a granular-ball rough concept tree (GBRCT). GBRCT enables GBRS to be a powerful mining tool that can process continuous data and realize feature selection, knowledge representation, and classification at the same time. Furthermore, according to the concept of GBRCT, we introduce a highly efficient algorithm known as GBNRS+, which is more consistent with traditional rough sets.

3) We introduce a new granular-ball generation algorithm [46], which improves the stability and efficiency of granular-ball generation and combines it with granular-ball NRS (GBNRS) and GBNRS+ for comparative experiments.

4) The experimental results demonstrate that the learning accuracy of GBRs has been significantly improved compared with the PRS and NRS. The GBRs also outperforms other seven popular or state-of-the-art feature selection methods.

The rest of this article is organized as follows. We introduce related works in Section II. The theory basis of GBRs is presented in Section III. Section IV details our newly proposed GBRS model, and the experimental results and analysis are presented in Section V. We present our conclusion in Section VI.

II. RELATED WORK

Since rough set is mainly used for feature selection, in this section, we present a more detailed discussion of prior works on feature selection methods as well as the rough set theory.

A. Filter Methods

They utilize statistical indicators to score features, such as the Pearson correlation coefficient, the Gini coefficient, the Kullback–Leibler divergence, the Fisher score, similarity measures, and so forth. Since filter methods only use the dataset itself and do not rely on specific classifiers, they are very versatile and are easy to expand. Compared with wrapper methods and embedded methods, a filter method usually possesses a lower algorithm complexity. At the same time, the classification accuracy of filter methods is usually the lowest among these three types of methods. Filter methods also only score a single feature rather than an entire feature subset. Thus, there can be a high redundancy in the feature subset generated by a filter method.

Gu et al. [47] proposed a generalized Fisher score feature selection method, aiming to find a feature subset that maximizes the lower bound of the Fisher score. This method transforms feature selection into a quadratically constrained linear program and uses a cutting plane algorithm to solve the problem. Roffo and Melzi [48] proposed a feature selection method based on graphs, which ranks the most important features based on recognition as arbitrary clue sets. This method maps feature selection to an affinity graph by assigning features as nodes and then evaluates the importance of each node via the eigenvector centrality. In a later work, Roffo et al. [49] proposed the Inf-FS feature selection method, which assigns features as nodes of a graph and views feature subsets as paths in the graph. The power series property of matrices is used to evaluate the path, and the computational complexity is reduced by adding paths until the length reaches infinity.

B. Wrapper Methods

They use learning algorithms to evaluate features. The classification accuracy of a wrapper method is often higher than that of a filter method. At the same time, the classifier used for evaluation limits the method, and the feature subset
obtained by wrapper methods tends to have lower versatility. For each feature subset, a wrapper method needs to train a classifier, resulting in a high computational complexity, which depends on the search strategy of the feature subset. However, wrapper methods evaluate the entire feature subset rather than a single feature by considering the dependence between features, so the redundancy of the resulting feature subset is often lower than that of a filter method.

The support vector machine (SVM) is a commonly used learning algorithm in wrapper methods. Guyon et al. [50] proposed a feature selection method using SVM in combination with recursive feature elimination. The method constructs the ranking coefficient of features according to the weight vector generated by the SVM during training. In each iteration, the feature with the smallest ranking coefficient is removed. Eventually, one can obtain a feature list sorted in descending order. Guo et al. [51] proposed a feature selection method based on clustering, which uses a triplet-based ordinal locality-preserving loss function to capture the local structures of the original data. The method defines an alternating optimization algorithm based on half-quadratic minimization to speed up the optimization process of this algorithm. Guo and Zhu [52] specifically developed another wrapper method, dependence-guided unsupervised feature selection (DGUFS), to address the issue of single feature selection in filtering methods using a joint learning framework for feature selection and clustering. DGUFS is a projection-free feature selection model based on $L_{2,0}$-norm equality constraints and two defined dependence-guided terms that increase the correlation between the original data, cluster labels, and the selected features.

C. Embedded Methods

They embed feature selection into the learning algorithm. The feature subset can be obtained when the training process of the learning algorithm has been completed. This type of method is overall similar to a filter method, but the score of each feature is determined through model training. The idea behind these methods is to select the features important to model training when determining the model. Embedded methods are a compromise between filter methods and wrapper methods. Compared with filter methods, embedded methods can achieve a higher classification accuracy. Meanwhile, compared with wrapper methods, embedded methods have a lower algorithm complexity and are less prone to overfitting issues.

Bradley and Mangasarian [53] proposed an embedded feature selection method based on concave minimization and SVM. This method finds a separation plane, which distinguishes two point sets in the $n$-dimensional feature space while using as few features as possible. This method not only minimizes the weighted sum of the distance between the incorrectly classified points and the boundary plane but also maximizes the distance between the two boundary planes of the separation plane. Embedded methods are often based on regression learning algorithms. Nie et al. [54] proposed an efficient and robust feature selection method using a loss function based on $L_{2,1}$-norms to remove outliers. This method adopts joint $L_{2,1}$-norm minimization on the loss function and regularization. In addition, the authors proposed an effective algorithm to solve joint $L_{2,1}$-norm minimization problems.

Yang et al. [55] proposed a feature selection method, unsupervised discriminative feature selection (UDFS), which also uses the $L_{2,1}$-norm. UDFS optimizes an $L_{2,1}$-norm regularized minimization loss function, which uses discriminative information and the local structure of the data distribution.

D. Rough Set Theory

Feature selection methods based on the rough set theory belong to filtering methods. These methods use the positive region from the rough set theory to score features. PRS granulates a dataset based on an equivalence relation, which provides good interpretability. This also results in that PRS can only process discrete data. Since most data in a real-world scenario are continuous, tremendous research has been poured into overcoming the inability of the rough set theory to process continuous data. This research can be roughly divided into two categories: discretizing continuous data or proposing improved rough set models. For decades, rough set models based on data discretization have proliferated [56], [57], [58]. However, the discretization of data will inevitably leads to the loss of information. In addition, the discretization results will change together with the discretization method. In light of this, some of them have proposed improved rough set models that can directly process continuous data. Dubois and Prade [60] combined rough sets with another concept, fuzzy sets [59], and proposed fuzzy rough sets, which replace the equivalence relation of classic rough sets with a fuzzy similarity relation so that fuzzy rough sets can process continuous data. On the other hand, fuzzy rough set models need to set a membership function in advance using prior knowledge of the dataset to reduce the generality of fuzzy rough sets.

In contrast to fuzzy rough sets, NRS [45] uses a neighborhood relation to describe the relationships between samples. This neighborhood relation is completely derived from the data distribution and does not require any prior knowledge. At the same time, NRS can process continuous data directly. Because of these advantages, the NRS has been under continuous study and development. Li and Xie [61] proposed a method to accelerate NRS based on an incremental attribute subset. Gao et al. [62] used a matrix to preserve measurement calculation results, which requires only one dimension measurement calculation after a dimension increase and thereby reduces the amount of calculations to find the positive region. In NRS, the neighborhood radius is a parameter that has a significant impact on the reduction results. Since this parameter must be artificially set up, how this parameter is chosen also becomes a popular topic. Peng et al. [63] designed a fitness function, which combines the properties of datasets and classifiers to select the optimal neighborhood radius from a given neighborhood radius interval. Xia et al. [64] proposed an adaptive NRS model by combining granular-ball computing with NRS, which can automatically optimize the neighborhood radius. Peng et al. [65] proposed a robust variable parameter granular-ball based on NRS for both attribute reduction and classification in label noise environments. The above NRS...
methods use a neighborhood relation instead of an equivalence relation to granulate datasets, thus enabling NRS to process continuous data directly. Although a series of theoretical models has been proposed in feature selection methods and rough set theory, they still face the following limitations and challenges.

1) In the model of NRS, the upper and lower approximations of the NRS consist of sample points instead of equivalence classes, so the NRS loses the interpretability.
2) The inconsistency between PRS and NRS makes the rough set theory not sufficiently concise.

In this article, we propose a novel rough set model named GBR, which unifies PRS and NRS. It not only has the interpretability of equivalence classes but can process continuous data naturally.

III. THEORY BASIS OF GBR

In this section, in order to lay a foundation for our theorem and proof, we review the basic concepts of PRS and NRS, which have been presented in our previous work [64]. We also introduce granular-ball computing, the basis of our proposed method, in this section.

A. Pawlak Rough Set

We first introduce the information system and the indiscernible relation.

Definition 1 (See [64]): Let a quaternion \( \langle U, A, V, f \rangle \) represent an information system where the following variables are defined.

1) \( U = \{x_1, x_2, \ldots, x_n\} \) denotes a nonempty finite set of objects. \( U \) is called the universe.
2) \( A = \{a_1, a_2, \ldots, a_m\} \) denotes a nonempty finite set of attributes.
3) \( V = \bigcup_{a \in A} V_a \) denotes the set of all attribute values, where \( V_a \) denotes the value range of attribute \( a \).
4) \( f = U \times A \to V \) denotes a mapping function: \( \forall x_i \in U, a \in A, f(x_i, a) \in V_a \).

This information system is called a decision system \( \langle U, C, D \rangle \) if the set of attributes in the information system above satisfies \( A = C \cup D, C \cap D = \emptyset \), and \( D \neq \emptyset \), where \( C \) is the condition attribute set and \( D \) is the decision attribute set.

Definition 2 (See [64]): Let \( \langle U, A, V, f \rangle \) be an information system. \( \forall x, y \in U \) and \( B \subseteq A \), the indiscernible relation \( \text{IND}(B) \) of the attribute subset \( B \) is defined as

\[
\text{IND}(B) = \{(x, y) \in U \times U | f(x, a) = f(y, a) \forall a \in B\}. \tag{1}
\]

In PRS algorithms, \( f(x, a) = V_a(x) \) represents \( x \)’s value on the attribute \( a \). Thus, \( f(x, a) = f(y, a) \) represents that the sample \( x \) has the same value as the sample \( y \) on the attribute \( a \). In fact, \( (x, y) \in \text{IND}(B) \) shows that the values of samples \( x \) and \( y \) are the same under the attribute subset \( B \), that is, under the description of the attribute subset \( B \), samples \( x \) and \( y \) are indiscernible.

\( \text{IND}(B) \) is symmetric, reflexive, and transitive; that is, \( \forall B \subseteq A \), \( \text{IND}(B) \) is an equivalence relation on \( U \) (abbreviated as \( R_B \)). \( \text{IND}(B) \) creates a partition of \( U \), denoted \( U/\text{IND}(B) \) and abbreviated as \( U/B \). The characteristics of \( U/B \) are given as follows. Suppose that \( U/B = \{X_1, X_2, \ldots, X_k\} \); if \( X_i, X_j \subseteq U, X_i \cap X_j = \emptyset(i \neq j) \), and \( \bigcup_{j=1}^k X_j = U \), then \( U \) is divided into \( k \) parts by \( \text{IND}(B) \). An element \( [x]_B = \{y \in U | (x, y) \in \text{IND}(B)\} \) in \( U/B \) is called an equivalence class. This inspires us to present the next set of definitions and approximations based on the equivalence relation \( R_B \).

Definition 3 (See [64]): Let \( \langle U, A, V, f \rangle \) be an information system. \( \forall B \subseteq A \), there is a corresponding equivalence relation \( R_B \) on \( U \). Then, \( \forall X \subseteq U \), the upper and lower approximation of \( X \) with respect to \( B \) are defined as follows:

\[
\overline{R_B}X = \{ y \in U/B | [x]_B \cap X \neq \emptyset \} \tag{2}
\]

\[
\underline{R_B}X = \{ y \in U/B | [x]_B \subseteq X \}. \tag{3}
\]

The lower approximation \( \underline{R_B}X \) represents the set of samples in \( U \) that are determined to belong to \( X \) according to the equivalence relation \( R_B \). It essentially reflects the ability of the equivalence relation \( R_B \) to approximately describe the knowledge contained in \( X \) by a partition of the knowledge of the universe \( U \). It is also commonly called the \( B \) positive region of \( X \) in \( U \), which is abbreviated as \( \text{POS}_B(X) \).

Definition 4 (See [64]): Let \( \langle U, C, D \rangle \) be a decision system. We notate the partition of the universe \( U \) by the decision attribute set \( D \) into \( L \) equivalence classes by \( U/D = \{X_1, X_2, \ldots, X_L\} \). \( \forall B \subseteq C \), there is a corresponding equivalence relation \( R_B \) on \( U \). The upper and the lower approximations of \( D \) with respect to \( B \) are, respectively, defined as

\[
\overline{R_B}D = \bigcup_{i=1}^L \overline{R_B}X_i \tag{4}
\]

\[
\underline{R_B}D = \bigcup_{i=1}^L \underline{R_B}X_i. \tag{5}
\]

Definition 5 (See [64]): Let \( \langle U, C, D \rangle \) be a decision system. \( \forall B \subseteq C \), the positive region and boundary region of \( D \) with respect to \( B \) are, respectively, defined as

\[
\text{POS}_B(D) = \overline{R_B}D \tag{6}
\]

\[
\text{BN}_B(D) = \overline{R_B}D - \underline{R_B}D. \tag{7}
\]

The size of the positive region reflects the separability of the classification problem in a given attribute space. The larger the positive region, the more detailed the classification problem can be described using this attribute set. We find it useful to describe this mathematically: the dependence of \( D \) on \( B \) is defined as

\[
\gamma_B(D) = \frac{|\text{POS}_B(D)|}{|U|} \tag{8}
\]

where \( |\cdot| \) is the cardinality of the set and \( 0 \leq \gamma_B(D) \leq 1 \). Obviously, the larger the positive region, the stronger the dependence of \( D \) on \( B \).

The dependence function defines the contribution of conditional attributes to a classification, so it can be used as an evaluation index for the importance of the attribute set.
Definition 6 (See [64]): Given a decision system \((U, C, D)\), \(\forall B \subseteq C\) and \(\forall a \in (C - B)\), the importance of \(a\) relative to \(B\) is defined as

\[
\text{SIG}(a, B, D) = \gamma_{B, \omega a}(D) - \gamma_B(D).
\] (9)

The rough set uses the measurement SIG in (9) to select attributes in a forward way. The selection result \(C'\) is initialized with \(\emptyset\), and for each attribute \(a\) in the attribute \(C - C'\), that with the largest value of \(\text{SIG}(a, C', D)\), which should be larger than 0, is selected into \(C'\). This process is repeated until all \(\text{SIG}(a, C', D)\) is not greater than 0.

B. Neighborhood Rough Set

After introducing NRS somewhat loosely, we now drill down into the details. We define the basic spaces and the neighborhoods in NRS. We also define the aforementioned positive region, which is key to the operation of these methods.

Definition 7 (See [64]): Let \(\Delta : \Omega \times \Omega \to R\) be a function generated on a set \(\Omega\). \((\Omega, \Delta)\) is known as a metric space if \(\Delta\) satisfies the following.

1) \(\Delta(x_1, x_2) \geq 0, \Delta(x_1, x_2) = 0\) if \(x_1 = x_2, \forall x_1, x_2 \in \Omega\).
2) \(\Delta(x_1, x_2) = \Delta(x_2, x_1), \forall x_1, x_2 \in \Omega\).
3) \(\Delta(x_1, x_3) \leq \Delta(x_1, x_2) + \Delta(x_2, x_3), \forall x_1, x_2, x_3 \in \Omega\).

In this case, \(\Delta\) is known as a metric.

Definition 8 (See [64]): Let \(U = \{x_1, x_2, \ldots, x_n\}\) be a nonempty finite set of real space. \(\forall x_i \in U\), the \(\delta\)-neighborhood of \(x_i\) is defined as

\[
\delta(x_i) = \{x | x \in U, \Delta(x, x_i) \leq \delta\}
\] (10)

where \(\delta \geq 0\).

Definition 9 (See [64]): Let \((U, C, D)\) be a neighborhood decision system. The decision attribute set \(D\) divides \(U\) into \(L\) equivalence classes: \(X_1, X_2, \ldots, X_L\). \(\forall B \subseteq C\), the lower approximation and the upper approximation of the decision attribute set \(D\) with respect to the condition attribute set \(B\) are, respectively, defined as

\[
N_B D = \bigcup_{i=1}^{L} N_B X_i
\] (11)

\[
\overline{N_B} D = \bigcup_{i=1}^{L} \overline{N_B} X_i
\] (12)

where \(N_B X_i = \{x_k | \delta_B(x_k) \subseteq X_i, x_k \in U\}\) and \(\overline{N_B} X_i = \{x_k | \delta_B(x_k) \bigcap X_i \neq \emptyset, x_k \in U\}\). Its positive region and boundary region are, respectively, defined as \(\text{POS}_D = \overline{N_B} D\) and \(\text{BN}(D) = \overline{N_B} D - N_B D\).

C. Granular-Ball Computing

Combining the theoretical basis of traditional granular computing, and based on the research results published by Chen [66] in science, he pointed out that “human cognition has the characteristics of large-scale priority,” and Wang [67] put forward a lot of granular cognitive computing. Based on granular cognitive computing, granular-ball computing is a new, efficient and robust granular computing method proposed by Xia et al. [68], the core idea of which is using “granular balls” to fully or partially cover the sample space. A granular ball \(GB = \{x_i, i = 1, \ldots, t\}\), where \(x_i\) represents the objects in GB and \(t\) is the number of objects in GB. GB’s center \(c\) and radius \(r\) are, respectively, represented as follows:

\[
c = \frac{1}{t} \sum_{i=1}^{t} x_i
\] (13)

\[
r = \max_{i=1}^{t} |x_i - c|.
\] (14)

This indicates that the radius is equal to the maximum distance from all objects in GB to its center. The radius can also be set to the average distance. The “granular ball” 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 and multiple scales) and the accurate characterization of the sample space.

Given a dataset \(D = \{x_i, i = 1, 2, \ldots, n\}\), where \(x_i\) and \(n\) represent the samples and the number of samples in \(D\), respectively, \(GB_j (j = 1, 2, \ldots, k)\) is a granular ball generated on \(D\) and \(k\) represents the total number of granular balls generated on \(D\). The granular-ball generation in granular-ball computing is mainly measured by the following factors.

1) In coverage degree, holding other factors constant, the higher the coverage degree, the less sample information is lost, and the representation is more accurate. Assuming that the number of samples in the \(j\)th granular ball is represented as \(|GB_j|\), the coverage degree can be expressed as: \((\sum_{j=1}^{k} |GB_j|)/n\). In the current granular-ball computation method, only in some cases of the classification problem, because the average radius is used in a granular ball, it is a partial coverage, and others, such as granular-ball graph representation and granular-ball clustering, are fully covered.

2) Holding other factors constant, the number of granular-balls is related to the coarseness of the balls. Minimizing this factor is to make the granular balls as coarse as possible, in line with the cognitive rule of “large scale first.” The fewer granular balls there are, the coarser the granularity, and the more corresponding coarse-grained characteristics are: granular-ball computing is more efficient, and the robustness is better.

3) Under different problems, to optimize the corresponding learning objectives, the quality of the granular balls themselves quality(GB) needs to be higher than a given threshold \(T\). This factor is also related to the lower limit of granular-ball coarseness, making the smallest granular-ball “fine” enough to accurately represent the problem. The threshold can be obtained through a given method, grid search, or adaptive method. The optimization objective function of granular-ball computing is shown as follows:

\[
\min \frac{n}{\sum_{j=1}^{k} |GB_j|} + k
\]

s.t. quality(GB) \(\geq T\). (15)
The basic process of granular-ball generation in granular-ball computing is shown in Fig. 1. To simulate “the characteristics of large-scale priority of human cognition” at the beginning of the algorithm, the whole dataset can be regarded as a granular ball. At this moment, the purity of the granular ball is the worst and cannot describe any distribution characteristics of the data. The “purity” is used to measure the quality of a granular-ball [68]. It is equal to the proportion of the most labels in the granular ball. Then, the number of labels $m$ of different classes in the granular ball is counted, and the granular ball can be split into $m$ granular balls. The next step is to calculate the purity of each granular ball. This is a key step because purity is the criterion for evaluating whether a granular ball needs to continue to split. As the splitting process continues to advance, the purity of the granular balls increases, and the decision boundary becomes increasingly clearer. When the purity of all granular balls meets the constraint conditions, the boundary becomes its clearest status and the algorithm converges. The final granular balls that no longer split are called the stable granular balls. The granular-ball computing has developed granular-ball classifiers [68], granular-ball clustering [69], GBRS [64], granular-ball sampling methods [70], and optimized granular-ball generation methods [71]. In addition, granular-ball computing can also be introduced into deep learning [72]. First, a graph representation based on granular balls is obtained. Furthermore, the image learning model is transformed from the convolutional neural network model to the graph neural network learning model. Recently, Xie et al. [46] proposed a new method of granular-ball generation. The core concept is that, when splitting each granular ball, we only need to calculate the distance from the center of samples with the same class label to the undivided samples instead of randomly selecting a center and computing its distance to all samples. This approach accelerates the speed of the granular-ball generation and enhances the stability.

IV. GRANULAR-BALL ROUGH SETS

A. Motivation

The main difference of upper or lower approximation between the model of PRS and NRS is that, as shown in Definitions 3 and 9, respectively, the former consists of equivalence classes, which can be used to represent knowledge and has the interpretability. However, the latter consists of sample points, which has no interpretability. If we want to use equivalence classes to describe the upper and lower approximation of NRS, a straightforward approach is to treat all objects in a neighborhood radius as an equivalence class. However, we find that this may make two equivalence classes with different decision labels equal. We called this phenomenon as “heterogeneous transmission.” It can be described in detail in Fig. 2. As shown in Fig. 2, according to the Definition 9, the objects, including $x_1$ and $x_3$–$x_5$, belong to the positive region, and $x_6$ belongs to the boundary region. The heterogeneous transmission appears in the intersecting area of the neighborhood area of $x_4$ and that of $x_5$. The intersecting area is called “transmission area.” When we define that the objects belong to a given neighborhood as an equivalence class, the label of the neighborhood equivalence class of $x_4$ is equal to “+$1$,” and the label of the neighborhood equivalence class of $x_5$ is equal to “$-1$.” However, a new object $x_6$ in transmission area is equivalent to the neighborhood equivalence class of $x_4$ and that of $x_5$ at the same time. This makes the two equivalence classes with two different label equivalents and obviously hinders the learning. To avoid the heterogeneous transmission phenomenon, one can set the neighborhood radius sufficiently small. However, this may make most of objects or all objects always belong to the positive region, while the positive region cannot be effectively used for measuring feature importance or other learning tasks. Overall, the heterogeneous transmission phenomenon is caused by the overlap between the positive region neighborhoods with different labels in NRS.

B. Granular-Ball Rough Set

GBRS is based on granular-ball computing, so it must follow (15). More narrowly, quality($GB_j$) and $T$ is replaced by purity($GB_j$) and the purity threshold $1$, respectively. In a rough set, all samples in the universe must be covered. Thus, the objective function of GBRS can be described as covering all samples with the minimum number of stable granular balls, which should meet two constraint conditions: 1) purity($GB_j$) = 1 and 2) no overlap between the heterogeneous granular-balls. Condition 1) guarantees the quality of each granular ball, and condition 2) can make the granular balls converge.
Fig. 2. Phenomena of the heterogeneous transmission. The label of a black circle point is equal to “+1,” and the label of a red plus point is equal to “−1.” The triangle point \(x_6\) is a new test sample.

Fig. 3. Phenomena of the heterogeneous transmission in Fig. 2 is removed using granular-ball computing.

fit the datasets well and eliminate the heterogeneous transmission phenomena mentioned in Fig. 2. If the two constraint conditions cannot be met simultaneously, condition 2) must be met at least. In the granular-ball generation of GBRS, the granular balls, which do not meet the constraints, need to be split by using the \(k\)-means clustering, whose default distance measurement function is Euclidean distance. Due to the fact that the Euclidean distance can directly process both discrete and continuous data, GBRS can also directly process both discrete and continuous data. As shown in Fig. 3, the neighborhood radii are adaptively different, and the overlap between the heterogeneous granular balls in the positive region does not exist in GBRS. Therefore, it is possible to use \(\{x\}_{GB(B)}\) for the relationship between \(x\) and \(y\) is denoted as \(x \sim y\).

In the GBRS, \(f(x, a) = GB, x \in GB\). Thus, \(f(x, a) = f(y, a) = GB\) represents that \(x\) and \(y\) belong to the same granular ball under the given attribute set \(a\). \(\forall B \subseteq A\), \(INDGB(B)\) is an equivalence relation on \(U\) (abbreviated as \(GBR_B\)). Because the granular balls do not overlap, \(GBIND(B)\) can also create a partition of \(U\), denoted \(U/GBIND(B)\) and abbreviated as \(U/GB(B)\). An element \([x]_{GB(B)} = \{y \in U | (x, y) \in INDGB(B)\}\) in \(U/GB(B)\) is an equivalence class generated by granular-ball computing.

Definition 12: Given an information system \((U, A, V, f)\), if \(GB_i \cap GB_j \neq \emptyset\), \(GB_i \sim GB_j\).

According to condition 2), there is no overlap between heterogeneous granular balls. If \(GB_i \cap GB_j \neq \emptyset\), the labels of \(GB_i\) and \(GB_j\) are the same, i.e., \(GB_i\) and \(GB_j\) belong to the same class. According to Definition 11, the samples in \(GB_i\) are indiscernible; the samples in \(GB_j\) are indiscernible, too. Because \(GB_i \cap GB_j \neq \emptyset\), \(GB_i\) and \(GB_j\) are indiscernible, that is, \(GB_i \sim GB_j\).

Since the overlap does not exist between the granular balls with different labels in GBRS, Definition 12 means that those granular balls with the same label belong to an equivalence class. This type of overlap between the positive region neighborhoods, i.e., granular balls, with the same label is not considered in this method because it does not lead to a heterogeneous transmission nor does it affect the decision.

1) Properties of GBRS: Given an information system \((U, A, V, f)\), \(x, y, z \in U, B \subseteq A\), \(\sim\) represents the indiscernible granular-ball relation of the attribute subset \(B\) on \(U\). The indiscernible granular-ball relation obviously has the following properties.

1) Symmetry: If \(x \sim y\), then \(y \sim x\).
2) Reflexivity: \(x \sim x\).
3) Transitivity: If \(x \sim y\) and \(y \sim z\), then \(x \sim z\).

In summary, similar to that in PRS, \(INDGB(B)\) is symmetric, reflexive, and transitive, and completely consistent with \(IND(B)\) in PRS.

Based on the equivalence class \([x]_{GB(B)}\), the definitions of positive region and upper and lower approximations are the same as those in PRS. Therefore, GBRS has a model consistent with that of PRS. Their specific definitions are shown as follows.

Definition 13: Let \((U, A, V, f)\) be an information system. \(\forall B \subseteq A\), there is a corresponding equivalence relation \(GBR_B\) on \(U\). Then, \(\forall X \subseteq U\), the upper and lower approximations of \(X\) with respect to \(B\) are defined as follows:

\[
\overline{GBR_B}X = \bigcup \{[x]_{B} \in U/GB(B) | [x]_{GB(B)} \cap X \neq \emptyset\} \quad (18)
\]

\[
\underline{GBR_B}X = \bigcup \{[x]_{B} \in U/GB(B) | [x]_{GB(B)} \subseteq X\} \quad (19)
\]
According to Definition 13, we can see that the upper and lower approximations of GBRS are composed of granular balls, i.e., equivalence classes. As a result, GBRS can be used to represent knowledge.

**Definition 14**: Let \((U, C, D)\) be a decision system. We notate the partition of the universe \(U\) by the decision attribute set \(D\) into \(L\) equivalence classes by \(U/D = \{X_1, X_2, \ldots, X_L\}\). \(\forall B \subseteq C\), there is a corresponding equivalence relation \(GBR_B\) on \(U\). The upper and the lower approximation of \(D\) with respect to \(B\) are, respectively, defined as

\[
GBR_B D = \bigcup_{i=1}^{L} GBR_B X_i (20)
\]

\[
GBR_B D = \bigcup_{i=1}^{L} GBR_B X_i . (21)
\]

According to Definition 14, a granular ball whose purity is equal to 1, i.e., that the samples in it have the same decision label, belongs to lower approximation (i.e., the positive region described in Definition 15) in a decision system.

**Definition 15**: Let \((U, C, D)\) be a decision system. \(\forall B \subseteq C\), the positive region and boundary region of \(D\) with respect to \(B\) are, respectively, defined as

\[
POS_B(D) = GBR_B D
\]

\[
BN_B(D) = GBR_B D - GBR_B D . (23)
\]

Completely the same as that in the PRS, the size of the positive region reflects the separability of the classification problem in a given attribute space. The larger the positive region, the more detailed the classification problem can be described using this attribute set. The dependence of \(D\) on \(B\) and \(SIG(a, B, D)\) is also the same as that in PRS.

In the models of GBRS, when the radius of each granular ball is set to an infinitely small positive number, GBRS is transformed into PRS. When the PRS algorithm is designed from the perspective of GBRS, it is called granular-ball PRS (GBPRS). GBPRS and PRS have the same experimental results; however, their algorithm designs are different, and the former algorithm generates equivalence classes using granular-ball computing. When the radius of each granular ball is not set to zero, GBRS is transformed into GBNRS. Since the GBNRS not only can use equivalence classes to represent knowledge but is also much more efficient than the traditional NRS, which contains many overlaps, the GBNRS can completely replace the traditional NRS. In other words, the GBNRS is the representative method of NRS. To summarize, GBRS is a unified model of GBPRS and GBNRS.

In addition, as shown in Fig. 3, GBNRS can flexibly fit the data distribution using those granular balls with various radii, which is obviously better than those methods using a fixed radius, such as PRS and the traditional NRS. Thus, GBNRS can achieve a higher accuracy than these two algorithms. Moreover, the combination of the robustness and adaptability of granular-ball computing is helpful to increase the accuracy of GBNRS. This robustness in the GBNRS is reflected in the fact that, since the noise point will be in the small granular ball, the characteristics of a large neighborhood, i.e., whether it belongs to a positive region or not, will not be affected. Such a robustness does not exist in other most methods, for example, the traditional NRS whose radius is fixed. GBRS can not only process both discrete and continuous data compared to PRS but also represent knowledge compared to NRS. We will further elaborate them in Section V.

An important advantage of GBNRS in this article is that it can overcome the overfitting in the classical rough set as described in [73]. For an extreme example, a noise attribute whose elements are integer sequence will make all samples belong to positive region and other attributes cannot be selected. From multigranularity perspective, it is because the granularity of neighborhood is fixed, i.e., the minimal value zero; on the contrary, the real multigranularity neighborhood of GBNRS can overcome the overfitting. This leads GBNRS in this article to a higher classification accuracy than other rough set methods, including the old GBNRS in [64].

### C. Implementation of GBNRS

Because the GBNRS has the unified model with the PRS, as shown in Fig. 4, its whole algorithm process is completely the same as that of the PRS. The only difference between the GBRS and PRS is the generation method of the positive region, which is shown in step 2 in Fig. 4. The GBRS generates positive regions using granular balls. In regard to the GBNRS, to fulfill Definition 14, the purity threshold \(PT\) is set to 1 in the granular-ball generation. Besides, according to [70], the lower bound of the size of a granular ball LBS, i.e., the number of samples in it, is optimized from \(2^d\) to 2 in a step size equal to 2. Here, \(d\) denotes the number of conditional attributes in the dataset. When the size of a granular ball is lower than LBS or its purity reaches 1, the granular ball stops splitting.

According to Definition 15, a granular ball whose purity is equal to 1 belongs to the positive region, while a granular ball whose purity is lower than 1 belongs to the boundary region. In order to decrease the randomness in the granular-ball generation and improve the comparability of the granular balls in the attribute selection process in the positive region, the samples with the smallest indexes are selected as the initial centroids in this granular-ball splitting process. As shown in Fig. 4, the attribute set whose elements are not selected \(C'\) and the reduction result \(C''\) are initialized with the entire attribute set \(C\) and \(\emptyset\), and \(pos_num = 0\). Subsequently, each attribute in \(C'\) is superimposed into \(C''\) in turn, and the algorithm in Fig. 1 is used to generate the balls. The attribute \(a_i\) is selected, where \(C'' = C'' \cup a_i\) have the most positive numbers named \(max_num\). If \(max_num > pos_num\), then attribute \(a_i\) is superimposed onto \(C''\), removed from \(C'\), and \(pos_num = max_num\). This process continues until the number of positive region samples no longer increases, that is, \(max_num \leq pos_num\). Furthermore, this article also introduces another novel and stable method of the granular-ball generation [46], which utilizes the center of the same label as the initial centroid. This method is included in the comparative experiment. As shown in Fig. 4, the process of attribute reduction in GBRS is composed of five major steps.
Fig. 4. Process of attribute reduction in GBRS.

Fig. 5. Generation process of GBNRS on the dataset. The red points and red granular balls are labeled “+1,” and the green points and green granular balls are labeled “−1.” First, the whole dataset is regarded as a granular ball, which is divided into two granular balls using two means because it contains two different classes of samples. Fig. 5(a)–(e) shows the intermediate iteration results. A granular ball with purity <1 and size ≥4, and it will continue splitting. Fig. 5(e) presents the phenomenon of heterogeneous transmission. This phenomenon is eliminated by splitting the heterogeneous overlapped granular balls to remove the overlap of heterogeneous balls. As shown in Fig. 5(f), any pair of two heterogeneous granular balls does not contain any common samples. In addition, those granular balls containing both green and red sample points, i.e., those black granular balls, belong to the boundary region.

D. GBRCT for Knowledge Representation and Classification

As the GBRS can use equivalence class to represent upper and lower approximations while processing continuous data, it presents advantages on representing the knowledge. In this section, we further propose the GBRCT by combining GBNRS with the rough concept tree (RCT) [73], which is proposed based on the concept lattice. The RCT can not only be used to organize and describe the knowledge rules obtained by rough sets based on the forward attribute reduction algorithm but also can be used for classification decision. Therefore, the GBRCT enables GBRS to be a strong mining tool that can realize feature selection, knowledge representation, and classification at the same time.
In the RCT, each node can be called a knowledge point or concept node. An RCT node consists of a sequence with the attributes/values called “intent” and its corresponding equivalence classes called “extent.” The concepts, “intent” and “extent,” are borrowed from the philosophy of knowledge representation. Since GBRS strictly divides a dataset using granular balls, for the GBRCT in the following, we suggest two parts in the first row and first column of a “knowledge point”: “the attribute value” and “the center and radius of the corresponding granular ball.”

In the GBRCT, different from that in the RCT, the intent is described using a granular-ball equivalence class consisting of its center and radius instead of a sequence of attribute values. The GBRCT generated on the discrete dataset zoo is shown in Fig. 6, and the GBRCT generated on the continuous dataset wine is shown in Fig. 7. Because we know that the dataset zoo is discrete in advance, the neighborhood radius that is smaller than an infinitely small positive value is used as the termination condition of granular-ball splitting, and GBRS is converted to GBPRS. In Figs. 6 and 7, an orange node represents a granular-ball equivalence class. A blue node containing “?” represents the boundary region, and an orange node representing a granular ball belongs to the positive region that can certainly describe knowledge. The dataset zoo is discrete such that the neighborhood radius is smaller than an infinitely small positive value, which is set to the termination condition of granular-ball splitting. The result of GBRCT is quit similar to the one using the RCT except the representation of the intent, i.e., the first row of each node. The RCT represents the intent using a sequence of attribute values, but the GBRCT uses a granular-ball consisting of a center and a neighborhood radius whose value is equal to zero, as shown in Fig. 6. On the contrary, as shown in Fig. 7, the neighborhood radius in the continuous data is larger than zero. This also indicates that the GBRS realizes the unified description for PRS and NRS well. In the real scenario, when the information on whether the dataset is discrete or not is not provided in advance, an infinitely positive value is considered as an option to be optimized as the neighborhood radius in the GBRS.

Similar to that in the RCT, the number in the right of a layer of the GBRCT shows the attributes that are generated on by the concept nodes in the layer. In addition, those positive region concept nodes containing the largest number of extent samples have the strongest representation ability for knowledge and are the most valuable, such as the second node in the second layer and the second node in the third layer in Fig. 6. Likewise, the GBNRS+ algorithm does not utilize the entire dataset but the boundary region to granular-ball splitting in feature selection. In fact, according to the theorem in [73], that is, the positive region of a subattribute set belongs to the positive region of the entire attribute set. This method speeds up calculations and is more consistent with traditional rough sets. Furthermore, the granular-ball acceleration method [46] is incorporated into GBNRS+, named the GBNRS ′ model. Moreover, as described in [73], the GBRCT can also be directly used for classification.

### E. Algorithm Design

The only difference between the GBNRS and PRS is the generation method of the positive region. The GBNRS and PRS share the same feature selection process because they both have a unified representation model. Therefore, we only discuss the algorithm design of granular-ball generation for the positive region in this section.
Algorithm 1 Generation of Granular Balls in the GBNRS

**Input:** A dataset \( D = \{x_1, x_2, \ldots, x_n\} \) based on the current attribute set, the lower bound of the size of the granular-ball \( LBS \);

**Output:** \( NOLGBs \);

1: The current granular-ball set \( CGBs \) and the granular-ball set in the next iteration \( NGBs \) are initialized with a granular-ball containing all sample points and \( \emptyset \);  // Generate the initial granular-balls.
2: repeat
3: for each granular-ball \( GB_i \in CGBs \) do
4: if \( purity(GB_i) < 1 \) and \( |GB_i| > LBS \) then
5: Split \( GB_i \) into \( k \) sub-granular-balls \( GB'_j, j = 1, \ldots, k \), where \( k \) denotes the number of different labels in \( GB_i \);
6: \( NGBs = NGBs + \{GB'_j\} \);
7: else
8: \( NGBs = NGBs + \{GB_i\} \);
9: end if
10: end for
11: \( CGBs = NGBs \); \( NGBs = \emptyset \);
12: until \( |NGBs| = |CGBs| \);  // If the granular-ball does not produce a new split, it will stop.
13: \( OLGBs = \emptyset ; NOLGBs = \emptyset \);
// Remove the overlap between heterogeneous granular-balls.
14: repeat
15: for each granular-ball \( GB \in CGBs \) do
16: if there is overlap between \( GB \) and \( GB \in CGBs \) which has a different label then
17: Split the larger granular-ball and add the subgranular-balls into \( OLGBs \);
18: else
19: \( NOLGBs = NOLGBs + GB \);
20: end if
21: end for
22: \( CGBs = OLGBs \)
23: until \( |OLGBs| = 0 \)

Algorithm 1 is the generation of granular balls in the GBNRS. As shown in Algorithm 1, the algorithm consists of two major stages: the initial granular-ball generation and the overlap removal. In the initial granular-ball generation, as shown in Algorithm 1 from steps 2 to 12, constraint condition 1) is used to control the splitting of the granular balls. In addition, the samples in each granular ball in this stage must be no less than \( LBS \). In the overlap removal, as shown in Algorithm 1 from steps 14 to 23, constraint condition 2) is used to remove the overlap of the granular balls with different labels.

In the process of initial granular-ball generation, we consider the total datasets as a granular ball first and put it in the current granular-ball set \( CGB_i \). Next, we will iterate through the granular ball in \( CGB \) and determine in sequence whether each granular ball meets the conditions in step 5. If so, we will use the \( k \)-means algorithm to split it into \( k \) subgranular balls. \( purity(GB) \) denotes the purity of the granular ball GB. At step 16, there is an overlap between two granular balls if their boundary distance is smaller than zero, i.e., the distance between their centers is smaller than the sum of their radii. The process of splitting at step 17 is similar to that at step 6. In the output variable \( NOLGBs \), the granular balls with a purity equal to 1 belong to the positive region. If a granular ball only has very few samples, such as only one sample, it cannot represent useful knowledge; if too many, it cannot represent fine knowledge and describe clear decision boundary. If there are granular balls during the splitting process that contain duplicate data from different labels, these balls do not meet the mass limit and cannot be further split. Consequently, we remove such balls. Referring to the suggestion in article [70], the lower bound of the size of a granular ball \( LBS \), i.e., the number of samples in it, is optimized from \( 2*d \) to 2 with a step size equal to 2. Here, \( d \) denotes the number of conditional attributes in the dataset.

The feature selection process based on GBNRS is consistent with Fig. 4, as shown in Algorithm 2. Actually, the boundary region consists of two types of balls. The first type includes small balls that fail to meet the purity requirements and in which the number of samples is less than the minimum number of samples. The second type consists of large balls that also do not meet the purity requirements but cannot be split. For instance, an existing ball contains three sample points: \( x_1 = (0, 1, 0) \), \( x_2 = (0, 0, 1) \), and \( x_3 = (0, 1, 1) \), each labeled as “+1,” “+2,” and “+3,” respectively. If the current
TABLE I
DATASET INFORMATION

| NO. | Dataset       | Size | NCA | CCA | Class |
|-----|---------------|------|-----|-----|-------|
| 1   | lymphography  | 148  | 0   | 18  | 4     |
| 2   | primary-tumor | 336  | 0   | 15  | 2     |
| 3   | mushroom      | 7535 | 0   | 22  | 2     |
| 4   | mushroom1     | 8124 | 0   | 22  | 2     |
| 5   | zoo           | 101  | 0   | 16  | 7     |
| 6   | backup-large  | 307  | 0   | 36  | 4     |
| 7   | ionic         | 351  | 34  | 0   | 2     |
| 8   | Diabetes      | 768  | 8   | 0   | 2     |
| 9   | wdbc          | 569  | 30  | 0   | 2     |
| 10  | audit Risk    | 772  | 21  | 0   | 2     |
| 11  | electrical    | 10000| 13  | 0   | 2     |
| 12  | Parkinson_Multiple_Sound_Recording | 1040 | 27 | 0 | 2 |
| 13  | wine          | 178  | 13  | 0   | 3     |
| 14  | spambase      | 4601 | 57  | 0   | 2     |
| 15  | htru2         | 17898| 8   | 0   | 2     |

V. EXPERIMENT

To demonstrate the feasibility and effectiveness of GBRS, we compare it against nine popular state-of-the-art algorithms on real-world datasets and add the results of GBNRS+, GBNRS′, and GBNRS′+ methods. Since the experimental results using GBPRS are the same as those using PRS, the GBNRS is selected for comparison. Considering that the PRS can only process discrete data, we also conduct experiments on discrete datasets when compared to PRS. As shown in Table I, we randomly select 15 real-world datasets, including continuous and discrete datasets to demonstrate the performance of the GBNRS. NCA denotes the numerical condition attributes. CCA denotes the categorical condition attributes. The first six datasets are discrete, while the last 11 ones are continuous. We also conducted a time comparison among the PRS, NRS, GBNRS, GBNRS+, GBNRS′, and GBNRS′+ methods. The latter four methods all rely on granular-ball computing, which involves the optimization of the minimum number of samples for each ball. In addition, the radius used in NRS is optimized within the range of $[0.01, 0.51]$ with a step size of 0.01. Therefore, the average time is selected as the benchmark for comparison. We carry out computer experiments on a PC equipped with an Intel Core i7-10700 CPU @2.90 GHz and 32-GB DRAM. The software is written using Python 3.7.

TABLE II
ACCURACY COMPARISON ON DISCRETE DATASETS

| NO. | Original       | PRS            | GBNRS          |
|-----|----------------|----------------|----------------|
| 1   | 0.8101±0.0966  | 0.7489±0.0627  | 0.8161±0.0584  |
| 2   | 0.6955±0.0290  | 0.6686±0.0414  | 0.6955±0.0290  |
| 3   | 0.9111±0.1034  | 0.9202±0.0753  | 0.9328±0.0746  |
| 4   | 1±0            | 0.9889±0.0026  | 1±0            |
| 5   | 0.95±0.0499    | 0.92±0.079     | 0.95±0.0353    |
| 6   | 0.9771±0.0274  | 0.9444±0.0526  | 0.9804±0.0269  |
| Avg | 0.8906         | 0.8618         | 0.8958         |

The lower bound of the size of a granular ball LBS, i.e., the unsupervised $k$-means method divides the three samples with different labels into a ball. Therefore, the balls that cannot be split and do not meet purity requirements are the boundary region. The difference from GBNRS is that GBNRS+ only performs granular-ball splitting on the boundary region. Since the positive region of a subattribute set belongs to the positive region of the entire attribute set, Therefore, the positive region obtained in the previous iteration does not need to participate in the next calculation. In fact, this approach greatly increases algorithm efficiency. In addition, the GBNRS′+ method based on the granular-ball acceleration method also uses this process.

A. In Comparison With PRS Under Discrete Data

The experimental results on the first six discrete datasets in Table I are shown in Table II, where the “original” column represents the classification accuracy obtained from the original unreduced dataset. The “NO” column in Table II is corresponds with the “NO” column in Table I. It can be seen from Table II that the classification accuracy of GBNRS is much higher than that of the PRS on most cases except the case on the 4th dataset, in which the accuracies of the two algorithms are the same. The original accuracy and PRS’s accuracy are 0.8906 and 0.8618, respectively, in average, while that of GBNRS is 0.8958. Compared with the previous two results, GBNRS achieves 0.52% and 3.4% enhancement. GBNRS obtains this superiority because it can flexibly fit the data distribution using those granular balls with various radii, which is better than those methods using a fixed radius, such as PRS and the traditional NRS. Therefore, GBNRS can achieve a higher accuracy than these two algorithms. In summary, the analysis shows that on discrete datasets, GBNRS can achieve a higher classification accuracy than both the PRS and those on the original datasets.

B. In Comparison With Various Feature Selection Methods

GBNRS can deal with both discrete and continuous datasets, so we use all datasets in Table I. We compare our model against nine popular state-of-the-art algorithms, including NRS [74], GBNRSold [64], Cfs [50], Ilfs [1], Laplacian [75], Lasso [76], Mrmr [22], WNRS [77], GBNRS′ [46], GBNRS+, and...
TABLE III
ACCURACY OF DIFFERENT ATTRIBUTE REDUCTION ALGORITHMS ON CONTINUOUS DATASETS

| Attribute         | NRS | GBNRS | GBNRS+ | GBNRS+ | GBNRS+ | GBNRS+ | GBNRS+ |
|-------------------|-----|-------|--------|--------|--------|--------|--------|
| 1                 | 0.9570 | 0.9260 | 0.9210 | 0.9235 |
| 2                 | 0.9660 | 0.9660 | 0.9660 | 0.9660 |
| 3                 | 0.9570 | 0.9260 | 0.9210 | 0.9235 |
| 4                 | 0.9660 | 0.9660 | 0.9660 | 0.9660 |
| 5                 | 0.9570 | 0.9260 | 0.9210 | 0.9235 |
| 6                 | 0.9660 | 0.9660 | 0.9660 | 0.9660 |
| 7                 | 0.9570 | 0.9260 | 0.9210 | 0.9235 |
| 8                 | 0.9660 | 0.9660 | 0.9660 | 0.9660 |
| 9                 | 0.9570 | 0.9260 | 0.9210 | 0.9235 |
| 10                | 0.9660 | 0.9660 | 0.9660 | 0.9660 |
| Avg               | 0.9570 | 0.9260 | 0.9210 | 0.9235 |

GBNRs+. The experimental results are shown in Table III. The neighborhood radius $\delta$ is gradually increased from 0.01 to 0.5 with a step size of 0.01, which is commonly used in the NRS and WNRS. The GBNRsold also introduces granular-ball computing to decrease the overlap in the traditional NRS, resulting in efficiency improvement. However, it does not realize the equivalence representation. Therefore, we named it with an “old” suffix, i.e., GBNRsold, to distinguish it from our method GBNRs. To amortize the randomness in the GBNRsold, we run GBNRs ten times and adopt the highest classification accuracy among the ten experimental results for comparison.

The experimental results are shown in Table III. In these comparison algorithms, Laplacian and Cfs have the lowest average accuracy, and GBNRs+ has the highest average result of 0.9235. GBNRs gets the second highest average accuracy. In fact, the accuracy of the last four methods in Table III varies very little. This is because they all combine the robustness and adaptability of granular-ball computing. The adaptability enables them to flexibly fit different data distributions using the granular balls with various radii, resulting in good accuracy. The accuracy average of GBNRsold is 0.9057, which is higher than GBNRs. In fact, since the positive region of a subattribute set belongs to the positive region of the entire attribute set, the model is more consistent with traditional rough sets. This method is more stable since it splits the boundary region during calculation. In addition, we also tested on high-dimensional datasets (leukemia and prostate), and the results are not satisfactory. The reason is that the data sample size is much smaller than the number of attributes, and it is meaningless for granular balls to describe these too sparse points.

Table IV shows the time results of PRS, NRS, GBNRs, GBNRs+, GBNRs− and GBNRs− models. GBNRs+ and GBNRs− process data only with the boundary region. Consequently, they are faster compared to GBNRs and GBNRs−. The new granular-ball generation method with fixed sample centers can generate balls faster and more stably, thus greatly speeding up calculation efficiency. Therefore, GBNRs− performs much better in speed than other methods on most datasets.

VI. CONCLUSION

This article presents a unified model for the two most popular rough set models, PRS and NRS models. The unified model can not only express knowledge with equivalence classes but also deal with both continuous data and discrete data. In comparison with nine popular state-of-the-art feature selection methods on 15 real-world datasets, we demonstrate that the proposed model can achieve better accuracy in most cases. However, the GBRS cannot achieve higher classification accuracy than the baselines in some cases. The reason may be that the quality of granular balls is not good enough. In short, we recommend using the GBNRs+ method, which is not only fast but also has better accuracy. In terms of theoretical applicability, granular-ball computing primarily relies on the multigranularity partitioning of the sample space. However, when the sample size is small, especially when the number of samples is close to the sample dimension, the sample distribution is already very sparse, and the distribution is approximately linearly separable in the classification. At this time, the further division of using granular-ball computing is of little significance. At present, granular-ball computing is not suitable for processing datasets with relatively similar dimensions and sample sizes. Besides, another future work will focus on extending our proposed model into other rough set models. In addition, we will also explore performance
optimization to accelerate the algorithm and improve the runtime efficiency.

ACKNOWLEDGMENT

The authors would like to thank Xiaoyao Liu and Yiping Xiong for their contributions on GBNRS and GBNRS+.

REFERENCES

[1] G. Roffo, S. Melzi, U. Castellani, and A. Vinciarelli, “Infinite latent feature selection: A probabilistic latent graph-based ranking approach,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 1407–1415.

[2] T. Y. Lin, “Granular computing: Practices, theories, and future directions,” Comput. Complex., vol. 2009, no. 1, pp. 4339–4355, 2009.

[3] D. B. Rawat, R. Doku, and M. Garuba, “Cybersecurity in big data era: From securing big data to data-driven security,” IEEE Trans. Services Comput., vol. 14, no. 6, pp. 2055–2072, Nov. 2021.

[4] D. Tripathi, D. R. Edla, and R. Cheruku, “Hybrid credit scoring model using neighborhood rough set and multi-layer ensemble classification,” J. Intell. Fuzzy Syst., vol. 34, no. 3, pp. 1543–1549, Mar. 2018.

[5] S. Luo, D. Miao, Z. Zhang, Y. Zhang, and S. Hu, “A neighborhood rough set model with nominal metric embedding,” Inf. Sci., vol. 520, pp. 373–388, May 2020.

[6] H. Yin, X. R. Li, and Y. Gao, “Relative Euclidean distance with application to TPSIS and estimation performance ranking,” IEEE Trans. Syst., Man, Cybern., Syst., vol. 52, no. 2, pp. 1052–1064, Feb. 2022.

[7] Z. Zhang, T. W. S. Chow, and M. Zhao, “Trace ratio optimization-based semi-supervised nonlinear dimensionality reduction for marginal manifold visualization,” IEEE Trans. Knowl. Data Eng., vol. 25, no. 5, pp. 1148–1161, May 2013.

[8] Y. Zhang et al., “Unsupervised nonnegative adaptive feature extraction for data representation,” IEEE Trans. Knowl. Data Eng., vol. 31, no. 12, pp. 2423–2440, Dec. 2019.

[9] Y. Zhang, D. Chen, and H. Wang, “Active sample selection based incremental algorithm for attribute reduction with rough sets,” IEEE Trans. Fuzzy Syst., vol. 25, no. 4, pp. 825–838, Aug. 2017.

[10] C. Wang, Q. Hu, X. Wang, D. Chen, Y. Qian, and Z. Dong, “Feature selection based on neighborhood discrimination index,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 7, pp. 2986–2999, Jul. 2018.

[11] P. Zhou, P. Li, S. Zhao, and X. Wu, “Feature interaction for streaming feature selection,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 10, pp. 4691–4702, Oct. 2021.

[12] Z. Zheng, G. An, S. Cao, D. Wu, and Q. Ruan, “Collaborative and multilevel feature selection network for action recognition,” IEEE Trans. Neural Netw. Learn. Syst., vol. 34, no. 3, pp. 1304–1318, Mar. 2023.

[13] M. Dash and H. Liu, “Feature selection for classification,” Intell. Data Anal., vol. 1, nos. 1–4, pp. 131–156, 1997.

[14] L. Sun, T. Yin, W. Ding, Y. Qian, and J. Xu, “Feature selection with missing labels using multilabel fuzzy neighborhood rough sets and maximum relevance minimum redundancy,” IEEE Trans. Fuzzy Syst., vol. 30, no. 5, pp. 1197–1211, May 2022.

[15] B. Sang, H. Chen, L. Yang, T. Li, W. Xu, and C. Luo, “Feature selection for dynamic interval-valued ordered data based on fuzzy dominance neighborhood rough set,” Knowl-Based Syst., vol. 227, Sep. 2021, Art. no. 107223.

[16] W. Li, H. Zhou, W. Xu, X.-Z. Wang, and W. Pedrycz, “Interval dominance-based feature selection for interval-valued ordered data,” IEEE Trans. Neural Netw. Learn. Syst., vol. 34, no. 10, pp. 6898–6912, Oct. 2023.

[17] P. Zhang, T. Li, Z. Yuan, C. Luo, K. Liu, and X. Yang, “Heterogeneous feature selection based on neighborhood combination entropy,” IEEE Trans. Neural Netw. Learn. Syst., early access, Aug. 4, 2022, doi: 10.1109/TNNLS.2022.3193929.

[18] C. Luo, S. Wang, T. Li, H. Chen, J. Lv, and Z. Yi, “Large-scale meta-heuristic feature selection based on BPSo assisted rough hyper-cubic approach,” IEEE Trans. Neural Netw. Learn. Syst., early access, May 12, 2022, doi: 10.1109/TNNLS.2022.3171614.

[19] Z. Pawlak, “Rough sets,” Int. J. Comput. Inf. Sci., vol. 11, no. 5, pp. 341–356, Oct. 1982.

[20] J.-M. Wei, S.-Q. Wang, and X.-J. Yuan, “Ensemble rough hypercubic approach for classifying cancers,” IEEE Trans. Knowl. Data Eng., vol. 29, no. 3, pp. 381–391, Mar. 2017.

[21] W. Ma, Y. Huang, H. Li, X. Li, and L. Jiao, “Image segmentation based on rough set and differential immune fuzzy clustering algorithm,” J. Softw., vol. 25, pp. 2675–2689, Jan. 2014.

[22] X. Qian, H. Xu, J. Liang, B. Liu, and J. Wang, “Fusing monotonic decision trees,” IEEE Trans. Knowl. Data Eng., vol. 27, no. 10, pp. 2717–2728, Oct. 2015.

[23] Y. Li, L. Zhang, Y. Xu, Y. Yao, R. Y. K. Lau, and Y. Wu, “Enhancing binary classification by modeling uncertain boundary in three-way decisions,” IEEE Trans. Knowl. Data Eng., vol. 29, no. 7, pp. 1438–1451, Jul. 2017.

[24] Y. Qian et al., “Local rough set: A solution to rough data analysis in big data,” Int. J. Approx. Reasoning, vol. 97, pp. 38–63, Jun. 2018.

[25] K. A. Vidyarthi and T. V. Geetha, “Entity resolution framework using rough set blocking for heterogeneous web of data,” J. Intell. Fuzzy Syst., vol. 34, no. 1, pp. 659–675, Jun. 2018.

[26] M. Hu and Y. Yao, “Structured approximations as a basis for three-way decisions in rough set theory,” Knowl-Based Syst., vol. 165, pp. 92–109, Feb. 2019.

[27] Y. Yao, “Decision-theoretic rough set models,” in Proc. Int. Conf. Rough Sets Knowl. Technol., 2007, pp. 1–12.

[28] W. Ziarko, “Variable precision rough set model,” J. Comput. Syst. Sci., vol. 46, no. 1, pp. 39–59, Feb. 1993.

[29] S. Parthalaïn, Q. Shen, and R. Jensen, “A distance measure approach to exploring the rough set boundary region for attribute reduction,” IEEE Trans. Knowl. Data Eng., vol. 22, no. 3, pp. 305–317, Mar. 2010.

[30] D. Chen, S. Zhao, L. Zhang, Y. Yang, and X. Zhang, “Sample pair selection for attribute reduction with rough set,” IEEE Trans. Knowl. Data Eng., vol. 24, no. 11, pp. 2080–2093, Nov. 2012.

[31] J. Liang, F. Wang, C. Deng, 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. 2014.

[32] H. Chen, T. Li, C. Luo, S.-J. Horng, and G. Wang, “A rough set-based method for updating decision rules on attribute values’ coarsening and refining,” IEEE Trans. Knowl. Data Eng., vol. 26, no. 12, pp. 2886–2899, Dec. 2014.

[33] P. Maji, “A rough hypercubic approach for feature selection in approximation spaces,” IEEE Trans. Knowl. Data Eng., vol. 26, no. 1, pp. 16–29, Jan. 2014.

[34] T. Y. Lin, “Neighborhood systems—A qualitative theory for fuzzy and rough sets,” in Advances in Machine Intelligence and Soft Computing, vol. 4, Jan. 1997, pp. 132–155.

[35] Y. Yao, “Rough sets, neighborhood systems and granular computing,” in Proc. Eng. Solutions Next Millennium, IEEE Can. Conf. Electr. Comput. Eng., Mar. 1999, pp. 1553–1558.

[36] T. Y. Lin, “Granular computing on binary relations I: Data mining and neighborhood systems,” Rough Sets Knowl. Discovery, vol. 1, no. 1, pp. 107–121, 1998.
[45] Q.-H. Hu, “Numerical attribute reduction based on neighborhood granulation and rough approximation,” J. Softw., vol. 19, no. 3, pp. 640–649, Jul. 2008.
[46] Q. Xie et al., “GBG++: A fast and stable granular ball generation method for classification,” 2023, arXiv:2305.18450.
[47] Q. Gu, Z. Li, and J. Han, “Generalized Fisher score for feature selection,” in Proc. 22th Conf. Uncertainty Artif. Intell., 2011, pp. 266–273.
[48] G. Roffo and S. Melzi, “Ranking to learn: Feature ranking and selection via eigenvector centrality,” in Proc. Int. Workshop New Frontiers Mining Complex Patterns, 2017, pp. 19–35.
[49] G. Roffo, S. Melzi, U. Castellani, A. Vinciarelli, and M. Cristani, “Infinite feature selection: A graph-based feature filtering approach,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 12, pp. 4396–4410, Dec. 2021.
[50] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” Mach. Learn., vol. 46, nos. 1–3, pp. 389–422, 2002.
[51] J. Guo, Y. Guo, X. Kong, and R. He, “Unsupervised feature selection with ordinal locality,” in Proc. IEEE Int. Conf. Multimedia Expo (ICME), Jul. 2017, pp. 1213–1218.
[52] J. Guo and W. Zhu, “Dependence guided unsupervised feature selection,” in Proc. 32th AAAI Conf. Artif. Intell., 2018, pp. 2232–2239.
[53] P. S. Bradley and O. L. Mangasarian, “Feature selection via concave minimization and support vector machines,” in Proc. ICML, vol. 98, Jul. 1998, pp. 82–90.
[54] F. Nie, H. Huang, X. Cai, and C. Ding, “Efficient and robust feature selection via joint $\ell_2,1$-norms minimization,” in Proc. Adv. Neural Inf. Process. Syst., vol. 23, 2010, pp. 1–9.
[55] Y. Yang, H. Shen, Z. Ma, Z. Huang, and X. Zhou, “$\ell_2,1$-norm regularized discriminative feature selection for unsupervised learning,” in Proc. 22th Int. Joint Conf. Artif. Intell., 2011.
[56] J. Liu, G. Wang, and F. Hu, “Cut’s discriminability based continuous attributes discretization in rough set,” J. Chongqing Univ. Posts Telecom., vol. 22, no. 4, pp. 257–261, 2010.
[57] R. Ali, M. H. Siddiqi, and S. Lee, “Rough set-based approaches for discretization: A compact review,” Artif. Intell. Rev., vol. 44, no. 2, pp. 235–263, Aug. 2015.
[58] F. Jiang and Y. Sui, “A novel approach for discretization of continuous attributes in rough set theory,” Knowl.-Based Syst., vol. 73, pp. 324–334, Jan. 2015.
[59] L. A. Zadeh, G. J. Klir, and B. Yuan, Fuzzy Sets, Fuzzy Logic, and Fuzzy Systems: Selected Papers, Singapore: World Scientific, 1996.
[60] D. Dubois and H. Prade, “Rough fuzzy sets and fuzzy rough sets,” Int. J. General Syst., vol. 17, nos. 2–3, pp. 191–209, 1990.
[61] N. Li and J. Y. Xie, “A feature subset selection algorithm based on neighborhood rough set for incremental updating datasets,” Comput. Technol. Develop., vol. 21, no. 11, pp. 149–155, 2011.
[62] Y. Gao, Z. Liu, and J. Ji, “Neighborhood rough set attribute reduction algorithm based on matrix reservation strategy,” Appl. Res. Comput., vol. 32, no. 12, pp. 3570–3573, 2019.
[63] X. Peng, Z. Liu, and J. Ji, “Adaptable method for determining neighborhood size of neighborhood rough set,” Appl. Res. Comput., vol. 36, no. 327, p. 4, 2019.
[64] S. Xia, H. Zhang, W. Li, G. Wang, E. Gien, 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. 2022.
[65] X. Peng, P. Wang, S. Xia, C. Wang, and W. Chen, “VPGB: A granular-ball based model for attribute reduction and classification with label noise,” Inf. Sci., vol. 611, pp. 504–521, Sep. 2022.
[66] L. Chen, “Topological structure in visual perception,” Science, vol. 218, no. 4573, pp. 699–700, Nov. 1982.
[67] G. Wang, “DGCC: Data-driven granular cognitive computing,” Granular Comput., vol. 2, no. 4, pp. 343–355, Dec. 2017.
[68] 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.
[69] S. Xia et al., “Ball k-means: Fast adaptive clustering with no bounds,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 1, pp. 87–99, Jan. 2022.
[70] 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., vol. 34, no. 4, pp. 2144–2155, Apr. 2023.
[71] S. Xia, X. Dai, G. Wang, X. Gao, and E. Gien, “An efficient and adaptive granular-ball generation method in classification problem,” IEEE Trans. Neural Netw. Learn. Syst., early access, Oct. 5, 2022, doi: 10.1109/TNNLS.2022.3203381.
[72] X. Shuyin et al., “Graph-based representation for image based on granular-ball,” 2023, arXiv:2303.02388.
[73] S. Xia et al., “An efficient and accurate rough set for feature selection, classification, and knowledge representation,” IEEE Trans. Knowl. Data Eng., vol. 35, no. 8, pp. 7725–7735, Aug. 2023.
[74] Q. Hu, D. Yu, J. Liu, and C. Wu, “Neighborhood rough set based heterogeneous feature subset selection,” Inf. Sci., vol. 178, no. 18, pp. 3577–3594, Sep. 2008.
[75] X. He, D. Cai, and P. Niyogi, “Laplacian score for feature selection,” in Proc. Adv. Neural Inf. Process. Syst., vol. 18, 2005, pp. 1–8.
[76] S. A. van de Geer, “High-dimensional generalized linear models and the lasso,” Ann. Statist., vol. 36, no. 2, pp. 614–645, Apr. 2008.
[77] M. Hu, E. C. C. Tsang, Y. Guo, D. Chen, and W. Xu, “A novel approach to attribute reduction based on weighted neighborhood rough sets,” Knowl.-Based Syst., vol. 220, May 2021, Art. no. 106908.