IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 24, 2019, accepted December 6, 2019, date of publication December 12, 2019, date of current version January 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2959126

Granule Vectors and Granular Convolutional Classifiers

YUMIN CHEN, XIAO ZHANG, WEI LI, AND SHUNZHI ZHU
College of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361024, China

Corresponding author: Yumin Chen (cym0620@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61976183 and Grant 61672442, in part by the Natural Science Foundation of Fujian Province under Grant 2019J01850 and Grant 2016J01325, and in part by the Social Science Planning Project of Fujian Province under Grant FJ2017C012.

ABSTRACT Convolutional operations can extract effective features and have been widely used in the field of deep learning. For the deficiency of convolution mainly dealing with numerical data, we propose a novel convolutional operator on granules with a set form, further we build a classifier on it. Firstly, feature granules are constructed on each single feature of a classification system by introducing neighborhood rough sets. Synchronously, decision granules are generated on the labels of samples. Secondly, feature granule vectors and weighted granule vectors are constructed from these granules, and a convolutional operation is proposed on feature granule vectors and weighted granule vectors, then a predicted granule is produced as a result of the convolutional operation. The predicted granule is compared with the decision granule, and their residual error is back propagated to the weighted granule vector for tuning its value. After multiple iterations of the granular convolutional operations and back propagation corrections, the weight of the granular vector is convergent and optimized. Furthermore, a granular classifier is designed based on the convolutional operation. The constringency of the granular convolution and the classification performance of the granular classifier are tested on some UCI datasets. Theoretical analysis and experimental results show that the granular convolution has a characteristic of fast convergence, and the granular convolutional classifier has a better classification performance.

INDEX TERMS Granular computing, neighborhood rough sets, convolutional network, granular classifier, rough sets.

I. INTRODUCTION

Convolution is an important operation in analytical mathematics. Let $f(x)$ and $g(x)$ be two integrable functions on R and their integral is: $\int_{-\infty}^{\infty} f(u)g(x-u)\,du$. It can be shown that for any $x \in (-\infty, \infty)$, its integral exists. Thus, with different values of x, the integral defines a new function $h(x)$, called the convolution of functions f and g, denoted as $h(x) = (f * g)(x)$. Convolution is widely used in mathematics [1], natural science [2], and practical engineering [3], [4]. The integer and polynomial multiplication in algebra and weighted moving average in statistics are convolutions. At the same time, convolution is also used in the fields of sound signal processing [5], digital communication [6] and bioinformatics [7], especially in the areas of image processing [8], [9], image segmentation [10], image denoising [11] and target recognition [12], [13].

The associate editor coordinating the review of this manuscript and approving it for publication was Fan-Hsun Tseng.

Convolutional operation is a common calculation method that has excellent performances in various fields. It has an advantage that the data reuse rate is high, which can greatly reduce the research cost. In the deep learning field, a convolutional network can accurately extract effective features. It achieves a great success in machine learning fields, such as face recognition [14], [15], object detection [16], video classification [17], natural language processing [18], etc [3], [19], [20]. Although, the theoretical foundation of deep learning has been investigated by some authors [21], [22], which includes convolutional operations, but the calculation of convolutional operations is highly complex. The convolutional network has huge parameters, resulting to a slow convergence and even a local solution. At the same time, convolution is mainly applicable to numerical data, but weakly handles sets with category data.

Rough set theory, proposed by Pawlak in 1982 [23], [24], is a mathematical tool for tackling set operations. Granular computing is originated from Zedeh [25] and Zedeh and

Lin [26], [27], which turns sets into granules. Yao proposed neighborhood systems and neighborhood granular computing [28], [29]. Pedrycz proposed a hyper box fuzzy granular classifier [30], [31]. Fujita discussed a systemic integration of granular computing from the perspective of cognition [32]. Liang and Qian analyzed uncertainty measurement in granular computing and its application in big data [33], [34]. Hu proposed a neighborhood granulation method for feature selection and classification [35]–[37]. We discuss granular structures, distances and measures [38], further proposed some operations on granules [39], [40]. Actually, these granules are sets. It is hard for convolutional operations running on these sets since they are discrete and non derivable. As for non-set data, there are many classification methods and systems including KNN [41], SVM [42], CNN [43], classifier ensemble [44], regression model [45], boosting and bagging systems [46], [47]. Since these classifiers focus on real-number data that are continue and derivable, the convolutional operation can be applied on them. In order to extend a function of convolution to set data, we construct some granule vectors and present a granular convolutional classifier in this paper. Firstly, a neighborhood rough set model is introduced for generating feature granules on each single feature in a classification system. As for the labels of samples, they are granulated into decision granules. Based on these granules, we define feature granule vectors and weighted granule vectors, and a convolutional operation is proposed on these vectors, then a new feature granule is formed as a result of the convolutional operation. Feature granules are compared with decision granules, and their residual errors are back propagated to the weighted granule vectors for tuning their values. Through the iterating process of granular convolutional operations and back propagation, the weight of granular vector is convergent and optimized. Furthermore, we design a granular classifier based on the convolutional operation. The constringency of granular convolution and the classification performance of a granular classifier are tested on some UCI datasets. Theoretical analysis and experimental results show that the granular convolution has a characteristic of fast convergence, and the granular convolutional classifier has a better classification performance.

The paper is structured as follows. First, we introduce the neighborhood granulation in Section II and present some granule vectors in Section III. Then, we propose a classification model based on granule vectors and granular convolution in Section IV. In Section V, we design an algorithm of granular convolutional classifier. In Section VI, we present experimental results. Conclusions and future works are covered in Section VII.

II. NEIGHBORHOOD GRANULATION

The rough set theory [23], proposed by Polish mathematician Pawlak, is one of the most widely used models in classification systems. In rough set theory, an equivalence class is considered as an elementary granule. For the numerical data that exists widely in the real world, the discretization process is needed. However, the discretization process is easy to cause the loss of classifying information. Therefore, Yao and Hu proposed a neighborhood rough set model [35]–[37] that is applied to the field of classification, while its neighborhood granulation is carried out from the whole features. In this paper, neighborhood granulation is performed on single features based on the Manhattan distance.

Suppose $CS = (S, F, L)$ is a classification system, where $S = \{x_1, x_2, \ldots, x_n\}$ is a set of samples or objects; $F = \{f_1, f_2, \ldots, f_m\}$ is a set of conditional features or attributes; and $L = \{l\}$ represents a decision attribute or label. The values of samples on feature set F are numerical, while their values on label set L are discrete or categories.

In a classification system, for any samples $x, y \in S$ and any single feature $a \in F$, the Manhattan distance between the samples x, y on the feature a is:

$$\Delta_a(x, y) = |v(x, a) - v(y, a)|, \tag{1}$$

where $v(x, a)$ represents the value of sample x on feature a.

In a classification system, let δ be a neighborhood granulation parameter, for any sample $x \in S$ and a single feature $a \in F$, the $\delta-$neighborhood granule of x on a is:

$$g_a^\delta(x) = \{y | x, y \in S, \Delta_a(x, y) \leq \delta\}. \tag{2}$$

For the decision label L, since its values are discrete, a sample can be granulated into a decision equivalent granule that is expressed as $g_L^0(x) = \{y | x, y \in S, \Delta_L(x, y) = 0\}$, which is the case where the neighborhood granulation parameter δ is equal to 0. When the granule is an empty set, it is called an empty granule that is marked as *null*; when the granule is a set of all samples, it is called a full granule that is noted as *full*.

According to the description of neighborhood granule, the δ-neighborhood granule of x on a is $g_a^\delta(x)$, which satisfies the following properties:

(1) $g_a^\delta(x) \neq \emptyset$;
(2) $x \in g_a^\delta(x)$;
(3) $y \in g_a^\delta(x) \Leftrightarrow x \in g_a^\delta(y)$;
(4) $\cup_{x \in S} g_a^\delta(x) = S$.

For a neighborhood granule $g_a^\delta(x)$ in the classification system, its size is expressed as:

$$Size(g_a^\delta(x)) = |g_a^\delta(x)|, \tag{3}$$

where $|.|$ represents the cardinality of a set. It is easy to know that the size of the neighborhood granule satisfies: $1 \leq Size(g_a^\delta(x)) \leq |S|$.

III. GRANULE VECTORS

Traditional vectors are quantities of both size and direction, so they are successfully applied in the machine learning field. The neighborhood granules are sets essentially, but sets have no vector representation. Therefore, the granules are difficult to be employed in the machine learning field. For solving this problem, we construct granule vectors in the follows.

Definition 1: Suppose a classification system is $CS = (S, F, L)$ and a neighborhood granulation parameter is δ, for any sample $x \in S$, any feature subset $P \subseteq F$, and

$P = \{a_1, a_2, \ldots, a_m\}$, then a $\delta-$neighborhood granule vector of x on the feature subset P is defined as:

$$vg_P^{\delta}(x) = (g_{a_1}^{\delta}(x), g_{a_2}^{\delta}(x), \ldots, g_{a_m}^{\delta}(x)), \tag{4}$$

where $g_a^{\delta}(x)$ is a $\delta-$neighborhood granule of the sample x on the feature a, which is called an element of the granule vector, referred to as a granule element. $vg_P^{\delta}(x)$ is a granule vector composed of granule elements. Therefore, the elements of the granule vector are sets. Unlike other vectors, the elements of a traditional vector are real numbers. When the elements of a granule vector are all empty granules, it is called an empty granule vector that is denoted as *vnull*; when the elements of a granule vector are full granules, it is called a full granule vector that is noted as *vfull*.

Definition 2: Suppose a classification system is $CS = (S, F, L)$ and a neighborhood granulation parameter is δ, for any sample $x \in S$, any feature subset $P \subseteq F$, and $P = \{a_1, a_2, \ldots, a_m\}$, then the size of a $\delta-$neighborhood granule vector $vg_P^{\delta}(x)$ of the sample x on the feature subset P is defined as:

$$|vg_P^{\delta}(x)| = \sqrt{\sum_{i=1}^{m} |g_{a_i}^{\delta}(x)|^2}. \tag{5}$$

The size of granule vector $vg_P^{\delta}(x)$ is also called the modulus of a granule vector.

Theorem 1: Let a classification system be $CS = (S, F, L)$, for any sample $x \in S$ and a single feature $a \in F$, suppose $g_a^{\gamma}(x)$, $g_a^{\delta}(x)$ are two neighborhood granules of x on a respectively, if $0 \le \gamma \le \delta \le 1$, then $g_a^{\gamma}(x) \subseteq g_a^{\delta}(x)$.

Proof: For $\forall x \in S$, according to the definition of neighborhood granule, there are $g_a^{\gamma}(x) = \{y | x, y \in S, \Delta_a(x, y) \le \gamma\}$ and $g_a^{\delta}(x) = \{y | x, y \in S, \Delta_a(x, y) \le \delta\}$. Because of $0 \le \gamma \le \delta \le 1$, it is easy to know $g_a^{\gamma}(x) \subseteq g_a^{\delta}(x)$.

Theorem 2: Let a classification system be $CS = (S, F, L)$, for any sample $x \in S$ and a feature subset $P \subseteq F$, suppose $vg_P^{\gamma}(x)$, $vg_P^{\delta}(x)$ are two neighborhood granule vectors of x on P respectively. If $0 \le \gamma \le \delta \le 1$, then $|vg_P^{\gamma}(x)| \le |vg_P^{\delta}(x)|$.

Proof: For $\forall a \in F$ and $0 \le \gamma \le \delta \le 1$, according to Theorem 1, then $g_a^{\gamma}(x) \subseteq g_a^{\delta}(x)$. Therefore, $|g_a^{\gamma}(x)| \le |g_a^{\delta}(x)|$ is established. For $\forall P \subseteq F$, according to Definition 2, we know that $|vg_P^{\delta}(x)| = \sqrt{\sum_{i=1}^{m} |g_{a_i}^{\delta}(x)|^2}$, $|vg_P^{\gamma}(x)| = \sqrt{\sum_{i=1}^{m} |g_{a_i}^{\gamma}(x)|^2}$. By $g_a^{\gamma}(x) \subseteq g_a^{\delta}(x)$, then $|vg_P^{\gamma}(x)| = \sqrt{\sum_{i=1}^{m} |g_{a_i}^{\gamma}(x)|^2} \le \sqrt{\sum_{i=1}^{m} |g_{a_i}^{\delta}(x)|^2} = |vg_P^{\delta}(x)|$. Therefore, $|vg_P^{\gamma}(x)| \le |vg_P^{\delta}(x)|$ is founded.

Example 1: A classification system $CS = (S, F, L)$ is shown in Table 1. Suppose $S = \{x_1, x_2, x_3, x_4\}$ is a sample set, $F = \{a, b, c\}$ is a feature set, and $L = \{l\}$ is a label set. The neighborhood granulation parameter is $\delta = 0.1$.

For the sample set $S = \{x_1, x_2, x_3, x_4\}$, if a neighborhood granulation is performed on feature a, the neighborhood granules are:

$$g_1 = g_a^{0.1}(x_1) = \{x_1, x_2\}, \quad g_2 = g_a^{0.1}(x_2) = \{x_1, x_2, x_3\},$$
$$g_3 = g_a^{0.1}(x_3) = \{x_2, x_3\} \text{ and } g_4 = g_a^{0.1}(x_4) = \{x_4\}.$$

TABLE 1. A classification system.

S	a	b	c	$\Rightarrow l$
x_1	0.1	0.2	0.1	1
x_2	0.2	0.5	0.2	1
x_3	0.3	0.3	0.3	0
x_4	0.7	0.1	0.3	0

If a neighborhood granulation is performed on feature b, the neighborhood granules are:

$$g_5 = g_b^{0.1}(x_1) = \{x_1, x_3, x_4\}, \quad g_6 = g_b^{0.1}(x_2) = \{x_2\},$$
$$g_7 = g_b^{0.1}(x_3) = \{x_1, x_3\} \text{ and } g_8 = g_b^{0.1}(x_4) = \{x_1, x_4\}.$$

If a neighborhood granulation is performed on feature c, the neighborhood granules are:

$$g_9 = g_c^{0.1}(x_1) = \{x_1, x_2\}, \quad g_{10} = g_c^{0.1}(x_2) = \{x_1, x_2, x_3, x_4\},$$
$$g_{11} = g_c^{0.1}(x_3) = \{x_2, x_3, x_4\} \text{ and } g_{12} = g_c^{0.1}(x_4) = \{x_2, x_3, x_4\}.$$

If a granulation is performed on label l, the decision equivalent granules are:

$$d_1 = g_l^0(x_1) = \{x_1, x_2\}, \quad d_2 = g_l^0(x_2) = \{x_1, x_2\},$$
$$d_3 = g_l^0(x_3) = \{x_3, x_4\} \text{ and } d_4 = g_l^0(x_4) = \{x_3, x_4\}.$$

If $P = \{a, b, c\}$, then the granule vector of x_1 on P is:

$$\begin{aligned} vg_P^{\delta}(x_1) &= (g_1, g_5, g_9) = (g_a^{0.1}(x_1), g_b^{0.1}(x_1), g_c^{0.1}(x_1)) \\ &= (\{x_1, x_2\}, \{x_1, x_3, x_4\}, \{x_1, x_2\}). \end{aligned}$$

The size of the granule vector is:

$$|vg_P^{\delta}(x_1)| = \sqrt{(2*2 + 3*3 + 2*2)} = 4.123.$$

Then the granule vector of x_2 on P is:

$$\begin{aligned} vg_P^{\delta}(x_2) &= (g_a^{0.1}(x_2), g_b^{0.1}(x_2), g_c^{0.1}(x_2)) \\ &= (\{x_1, x_2, x_3\}, \{x_2\}, \{x_1, x_2, x_3, x_4\}). \end{aligned}$$

The size of the granule vector is:

$$|vg_P^{\delta}(x_2)| = \sqrt{(3*3 + 1*1 + 4*4)} = 5.099.$$

Further the granule vector of x_3 on P is:

$$\begin{aligned} vg_P^{\delta}(x_3) &= (g_a^{0.1}(x_3), g_b^{0.1}(x_3), g_c^{0.1}(x_3)) \\ &= (\{x_2, x_3\}, \{x_1, x_3\}, \{x_2, x_3, x_4\}). \end{aligned}$$

The size of the granule vector is:

$$|vg_P^{\delta}(x_3)| = \sqrt{(2*2 + 2*2 + 3*3)} = 4.123.$$

And the granule vector of x_4 on P is:

$$\begin{aligned} vg_P^{\delta}(x_4) &= (g_a^{0.1}(x_4), g_b^{0.1}(x_4), g_c^{0.1}(x_4)) \\ &= (\{x_4\}, \{x_1, x_4\}, \{x_2, x_3, x_4\}). \end{aligned}$$

The size of the granule vector is:

$$|vg_P^{\delta}(x_4)| = \sqrt{(1*1 + 2*2 + 3*3)} = 3.742.$$

IV. THE GRANULAR CONVOLUTIONAL MODEL

We use neighborhood relations to granulate samples into neighborhood granules according to different features in a classification system, and these neighborhood granules construct a granule vector. At the same time, the samples according to their labels can also be granulated into decision granules by an equivalence relation. Vectors in traditional convolutional operations are real number-based operations and are not suitable for operations with a set form. The neighborhood granule is a form of set. Firstly, we present some set operations for granules and granule vectors. Then we propose a model of granular convolution based on these operations.

A. OPERATIONS OF GRANULES AND GRANULE VECTORS

Suppose a classification system is $CS = (S, F, L)$ and a neighborhood granulation parameter is δ. For any two samples $x, y \in S$, a single feature $a \in F$, let $s = g_a^\delta(x)$, $t = g_a^\delta(y)$ be two neighborhood granules of x, y on a, the union, intersection, subtraction and *XOR* operations of the two granules are represented as follows:

$$s \cup t = g_a^\delta(x) \cup g_a^\delta(y). \tag{6}$$

$$s \cap t = g_a^\delta(x) \cap g_a^\delta(y). \tag{7}$$

$$s - t = g_a^\delta(x) - g_a^\delta(y). \tag{8}$$

$$s \oplus t = g_a^\delta(x) \cup g_a^\delta(y) - g_a^\delta(x) \cap g_a^\delta(y). \tag{9}$$

Since sets of samples constitute neighborhood granules, the operations of intersection, union, subtraction and *XOR* between neighborhood granules are those of intersection, union, subtraction and *XOR* between sets. The result of *XOR* operation also represents the distance between two neighborhood granules.

In a classification system $CS = (S, F, L)$, for any two samples $x, y \in S$, there exists two $\delta-$neighborhood granule vectors $vg_F^\delta(x) = (g_{a_1}^\delta(x), g_{a_2}^\delta(x), \ldots, g_{a_m}^\delta(x))$, $vg_F^\delta(y) = (g_{a_1}^\delta(y), g_{a_2}^\delta(y), \ldots, g_{a_m}^\delta(y))$ on F, then the union, intersection, subtraction and *XOR* operations of the two granule vectors are represented as follows:

$$\begin{aligned} &vg_F^\delta(x) \cup vg_F^\delta(y) \\ &= (g_{a_1}^\delta(x) \cup g_{a_1}^\delta(y), g_{a_2}^\delta(x) \cup g_{a_2}^\delta(y), \ldots, g_{a_m}^\delta(x) \cup g_{a_m}^\delta(y)). \end{aligned} \tag{10}$$

$$\begin{aligned} &vg_F^\delta(x) \cap vg_F^\delta(y) \\ &= (g_{a_1}^\delta(x) \cap g_{a_1}^\delta(y), g_{a_2}^\delta(x) \cap g_{a_2}^\delta(y), \ldots, g_{a_m}^\delta(x) \cap g_{a_m}^\delta(y)). \end{aligned} \tag{11}$$

$$\begin{aligned} &vg_F^\delta(x) - vg_F^\delta(y) \\ &= (g_{a_1}^\delta(x) - g_{a_1}^\delta(y), g_{a_2}^\delta(x) - g_{a_2}^\delta(y), \ldots, g_{a_m}^\delta(x) - g_{a_m}^\delta(y)). \end{aligned} \tag{12}$$

$$\begin{aligned} &vg_F^\delta(x) \oplus vg_F^\delta(y) \\ &= (g_{a_1}^\delta(x) \oplus g_{a_1}^\delta(y), g_{a_2}^\delta(x) \oplus g_{a_2}^\delta(y), \ldots, g_{a_m}^\delta(x) \oplus g_{a_m}^\delta(y)). \end{aligned} \tag{13}$$

A granule vector is an ordered sequence. The operational result of two granule vectors is a granule vector that is also an ordered sequence.

B. GRANULAR CONVOLUTION

A classification system is $CS = (S, F, L)$, where a sample set is $S = \{x_1, x_2, \ldots, x_n\}$, and a feature set is $F = \{a_1, a_2, \ldots, a_m\}$. For $\forall x \in S$, an extended $\delta-$neighborhood granule vector of x on F is $vg_F^\delta(x) = (g_{a_1}^\delta(x), g_{a_2}^\delta(x), \ldots, g_{a_m}^\delta(x), full)$. Suppose a weighted granule vector shared by feature sets is $vq_F = (w_{a_1}, w_{a_2}, \ldots, w_{a_m}, b)$, the convolutional operation of two granule vectors is expressed as:

$$\begin{aligned} g_{F,w}^\delta(x) = vg_F^\delta(x) \bullet vq_F &= (g_{a_1}^\delta(x) \cap w_{a_1}) \\ &\cup (g_{a_2}^\delta(x) \cap w_{a_2}) \cup (\ldots) \cup (g_{a_m}^\delta(x) \cap w_{a_m}) \cup (full \cap b). \end{aligned} \tag{14}$$

For a traditional convolutional operation, the convolutional result of two vectors is a real scalar. Granular convolution also has this characteristic that the convolutional result of two granule vectors is a granular scalar. It induces a new granule named a predicted granule, since two granule vectors are converted into a granular scalar by a granular convolutional operation. This indicates that the granular convolution also has the function of feature extraction. At the same time, samples with category labels in a classification system can be granulated into granular scalars called decision granules. Therefore, two granules (a predicted granule and a decision granule) can be measured, and the comparison result is propagated back to the weighted granule vector, then the shared weighted granule vector is modified, so as to achieve more accurate results of feature extraction and classification. The process of granular convolution and back propagation is illustrated in Figure 1.

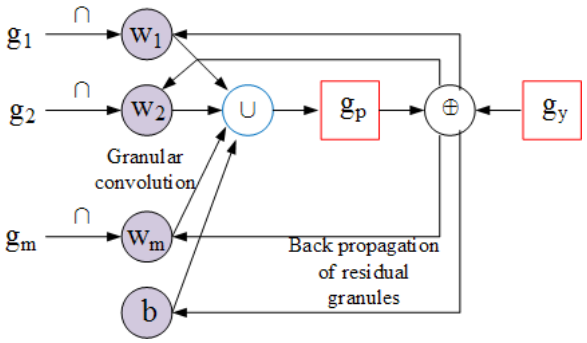


FIGURE 1. The process of granular convolution and back propagation.

In a classification system $CS = (S, F, L)$, for $\forall x \in S$, an extended $\delta-$neighborhood granule vector of x on F is $vg_F^\delta(x) = (g_{a_1}^\delta(x), g_{a_2}^\delta(x), \ldots, g_{a_m}^\delta(x), full)$. Suppose a weighted granule vector shared by feature sets is $vq_F = (w_{a_1}, w_{a_2}, \ldots, w_{a_m}, b)$, then the convolution of granule vector of x and weighted granule vector forms a predicted granule that is $g_{F,w}^\delta(x)$. And the sample x according to its label is granulated into a decision granule that is $g_L^0(x)$, then the residual errors between the predicted granule and the decision granule, named as positive and negative residual granules, are expressed as:

$$\begin{aligned} g^+(x) &= g_{F,w}^\delta(x) - g_L^0(x); \\ g^-(x) &= g_L^0(x) - g_{F,w}^\delta(x). \end{aligned} \tag{15}$$

C. BACK PROPAGATION OF RESIDUAL GRANULES

The convolutional operation of a feature granule vector and a weighted granule vector forms a predicted feature granule, and the measure between the predicted granule and a decision granule induces positive and negative residual granules. How to propagate these residual granules for correcting the weighted granule vector? It is a critical problem in granular classifiers. Therefore, we define some back propagation rules for residual granules in the follows.

Suppose a classification system is $CS = (S, F, L)$, where a sample set is $S = \{x_1, x_2, \ldots, x_n\}$, and a feature set is $F = \{a_1, a_2, \ldots, a_m\}$. For $\forall x \in S$, an extended $\delta-$neighborhood granule vector of x on F is $vg_F^{\delta}(x) = (g_{a_1}^{\delta}(x), g_{a_2}^{\delta}(x), \ldots, g_{a_m}^{\delta}(x), full)$, and a weighted granule vector is $vq_F = (w_{a_1}, w_{a_2}, \ldots, w_{a_m}, b)$. Let a positive residual granule of x be $g^+(x)$, and a learning rate is r, then some back propagation rules about the positive residual granule are:

1. $p = |g^+(x)| * r$;
2. Randomly select p granule elements from $g^+(x)$ to form a new positive residual granule that is $e^+(x) = \{x_1, x_2, \ldots, x_p\}$;
3. Loop from a_1 to a_m: If $e^+(x) \cap g_{a_i}^{\delta}(x) \cap w_{a_i}$ is not null, then $w_{a_i} = w_{a_i} - e^+(x) \cap g_{a_i}^{\delta}(x)$;
4. $b = b - e^+(x)$.

Let a negative residual granule of x be $g^-(x)$, and a learning rate is r, then some back propagation rules about the negative residual granule are:

1. $p = |g^-(x)| * r$;
2. Randomly select p granule elements from $g^-(x)$ to form a new positive residual granule that is $e^-(x) = \{x_1, x_2, \ldots, x_p\}$;
3. Loop from a_1 to a_m: If $e^-(x) \cap g_{a_i}^{\delta}(x)$ is not null, then $w_{a_i} = w_{a_i} \cup (e^-(x) \cap g_{a_i}^{\delta}(x)), e^-(x) = e^-(x) - e^-(x) \cap g_{a_i}^{\delta}(x)$;
4. $b = b \cup e^-(x)$.

Rule 1 and Rule 2 induce new residual granules based on a learning rate, and the learning rate is generally small, with values between 0 and 0.05. Rule 3 corrects w_{a_1} of the weighted granule vector vq_F, and Rule 4 corrects b of the weighted granule vector vq_F.

V. GRANULAR CONVOLUTIONAL CLASSIFIERS

The traditional classification models in machine learning are regression and classification, which mainly deal with real numbers. In this paper, both granules and granule vectors are forms of sets. We propose a granular convolutional classifier, which is a form of set operations, including granulation, granular learning and granular classification. The principle of granular convolutional classifiers is proposed, and a specific granular convolutional learning and a classification algorithm are designed.

A. PRINCIPLE OF GRANULAR CONVOLUTIONAL CLASSIFIERS

The granular convolutional classifier involves granulation, granular convolutional training, and granular classification. The granulation process includes three steps: data preprocessing, dividing training and test sets, and granulating the training set into granule vectors. The training process includes: initialization of weighted granule vector, convolutional operations on granule vectors, measurement of decision granules, back propagation of residual granules and correction of the weighted granule vector. The classification process includes: granulating the test set into granule vectors, convolutional operations on test granule vectors, and determining the labels of test granules. The granulation, training and classification processes are described in detail as follows.

1. Data preprocessing: Deleting data with missing values and normalizing the dataset between $0 \sim 1$.
2. Divided into training and test sets: The training set is 80% and the test set is 20%.
3. A granulation for the training set: According to a neighborhood parameter and single features, transforming each training sample into a feature granule vector; meanwhile, labels are granulated into a decision granule set.
4. Weight initialization: The weighted granule vector is randomly initialized.
5. Convolutional operation of granule vectors: The feature granule vector is convoluted with the weighted granule vector to form a new feature granule.
6. Measurement of residual errors: Feature granules are compared with decision granules to induce positive and negative residual granules.
7. Back propagation and weight correction: According to the back propagation rules, the weighted granule vector is updated by a learning rate.
8. Go to Step 5 and loop: Iteratively multiple times until the residual errors are converged.
9. A granulation for a test set: Take a test sample, compute the distance between the test sample and each training sample on a single feature, then obtain a test granule vector, and granulate all the test samples into test granule vectors.
10. Convolutional operation for a test granule vector: A test granule vector is convoluted with the previously trained weight vector to achieve a new feature granule.
11. The label judgment of a test granule vector: Comparing the distance between the feature granule and each decision granule, then the test granule vector is determined to be the class with the minimum distance.
12. Classification of next test granule vectors: Go to Step 10 and perform the classification of the next test granule vector until all test granule vectors are classified.

From the above analysis, a granular convolutional classifier is mainly divided into two processes: a training process and a test process. The training process is iterated until the residual errors are converged, so that the weighted granule vector revised in the training process is suitable for a subsequent granular classifying.

B. GRANULAR CONVOLUTIONAL TRAINING AND CLASSIFICATION ALGORITHMS

We give the principle and steps of a granular convolutional classifier in the above subsection. The granular convolutional

Algorithm 1 Granular Convolutional Learning (GCL)

Input: a training set $CS = (Tr, F, L)$, a neighborhood parameter δ and a learning rate r.

Output: a weighted granule vector $vq_F = (w_{a_1}, w_{a_2}, \ldots, w_{a_m}, b)$.

(1) Normalization for a training set: $Tr \in [0, 1]$;

(2) Initialization for the learning rate r and the weighted granule vector $vq_F = (w_{a_1}, w_{a_2}, \ldots, w_{a_m}, b)$;

(3) Iterate s times from Step (4) to Step (13):

(4) The residual error is assigned an initial value: $e = 0$;

(5) Perform Steps (6)-(12) for each training sample $x \in Tr$;

(6) According to the neighborhood parameter δ and each single feature $a_i \in F$, we granulate sample x into a feature granule $g^{\delta}_{a_i}(x)$;

(7) Form an extended $\delta-$neighborhood granule vector $vg^{\delta}_F(x) = (g^{\delta}_{a_1}(x), g^{\delta}_{a_2}(x), \ldots, g^{\delta}_{a_m}(x), full)$ of x;

(8) The label of x is granulated into a decision granule $d^0_L(x)$;

(9) A convolutional operation of feature granule vector and weighted granule vector induces a new feature granule: $g^{\delta}_{F,w}(x) = vg^{\delta}_F(x) \bullet vq_F$;

(10) Make a measurement between feature granule $g^{\delta}_{F,w}(x)$ and decision granule $d^0_L(x)$ to form a positive residual granule $e^+(x)$ and a negative residual granule $e^-(x)$;

(11) Modify the weighted granule vector vq_F according to the back propagation rules;

(12) Cumulate residual errors $e = e + |e^+(x)| + |e^-(x)|$;

(13) Exit the iteration if the residual error ratio e/e_{max} converges;

(14) Output weighted granule vector $vq_F = (w_{a_1}, w_{a_2}, \ldots, w_{a_m}, b)$.

Algorithm 2 Granular Convolutional Classification (GCC)

Input: a training set $CS = (Tr, F, L)$, a test sample t, a neighborhood parameter δ, a weighted granule vector $vq_F = (w_{a_1}, w_{a_2}, \ldots, w_{a_m}, b)$.

Output: the label of a test sample *label*.

(1) Normalization for the training set and the test sample: $Tr \in [0, 1]$, $t \in [0, 1]$;

(2) According to the neighborhood parameter δ and the training set, the test sample is granulated into $g^{\delta}_{a_i}(x)$;

(3) Form an extended neighborhood granule vector $vg^{\delta}_F(t) = (g^{\delta}_{a_1}(t), g^{\delta}_{a_2}(t), \ldots, g^{\delta}_{a_m}(t), full)$;

(4) A convolutional operation of feature granule vector and weighted granule vector forms a new feature granule $g^{\delta}_{F,w}(t) = vg^{\delta}_F(t) \bullet vq_F$;

(5) The m labels are granulated into decision granules $d_1, d_2, \ldots, d_m$ in the training set;

(6) Compute the distance between the feature granule $g^{\delta}_{F,w}(t)$ and each decision granule d_i: $dis_i(t) = |g^{\delta}_{F,w}(t) \oplus d_i|$;

(7) Determine the label of the test sample t as that of decision granule with the minimum $dis_i(t)$;

(8) Output the *label*.

classifier can be designed into two parts: the Granular Convolutional Learning (GCL) algorithm (training) and the Granular Convolutional Classification (GCC) algorithm. The details are described as follows.

In the algorithm GCL, it mainly involves the process of neighborhood granulation. In Step (6), hash sorting algorithm [48] is used for the neighborhood granulation of training set. Its time complexity is $O(m * n)$, where m is the number of features and n is the number of training samples; in Step (7), the time complexity is $O(m)$; in Steps (8), (10), the time complexity is $O(n)$; and in Steps (9), (11), the time complexity is $O(m * n)$. Since they are n loops in Steps (6)-(12), so the time complexity for them is $O(m * n^2)$. It iterates s times from Step (4) to Step (13). Therefore, in the worst case, the time complexity of GCL algorithm is $O(s * m * n^2)$. As for the GCC algorithm, its time complexity is $O(m * n)$.

VI. EXPERIMENTAL ANALYSIS

In this paper, Iris and Wine from the UCI dataset are used for experimental tests, and detailed results are made from neighborhood parameters, learning rate, convergence and classification accuracy. Due to the different value ranges of datasets, the datasets need to be normalized. We employ the maximum and minimum method to ensure that all data are converted to values between [0, 1]. The maximum and minimum normalization formula is:

$$f(x_i) = \frac{x_i - x_{min}}{x_{max} - x_{min}}. \tag{16}$$

The datasets are randomly divided into two parts according to 80% training samples and 20% test samples. The training process of a granular classifier has a random initialization. Each test result may be different. The test is performed 10 times under a same parameter, and the classification accuracy is the average of the 10 results.

A. INFLUENCES OF NEIGHBORHOOD PARAMETER

A dataset is granulated on each single feature by a neighborhood parameter to form a granule vector. Some experiments in this subsection mainly test the influences of varied neighborhood parameters. In these experiments, the neighborhood granulation parameters start from 0.05 to 0.5 with an interval of 0.05. For comparison purposes, three different learning rates are used, with Iris dataset at 0.06, 0.065, and 0.07, and Wine dataset at 0.02, 0.025, and 0.03. The experimental results are shown in Tables 2-3 and Figures 2-3. The horizontal axis is the neighborhood parameter and the vertical axis represents the classification accuracy.

It can be seen from Table 2 and Figure 2 that for Iris dataset, the classification accuracy is better while the neighborhood parameters are 0.1 and 0.15. When the learning rate is 0.06, the classification accuracy is better than that at the learning rates with 0.065 and 0.07. Further, the learning rate

TABLE 2. Influences of neighborhood parameter for Iris dataset.

$Neighborhood\_parameter$	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
$Learning\_rate = 0.060$	0.9067	0.9467	0.9600	0.9367	0.8767	0.7433	0.6633	0.6567	0.4433	0.3333
$Learning\_rate = 0.065$	0.8833	0.9533	0.9533	0.9333	0.8033	0.6900	0.6667	0.6200	0.4100	0.3333
$Learning\_rate = 0.070$	0.8667	0.9500	0.9467	0.8900	0.8133	0.6700	0.6633	0.6167	0.3633	0.3333

TABLE 3. Influences of neighborhood parameter for Wine dataset.

$Neighborhood\_parameter$	0.05	0.10	0.15	0.20	0.25	0.30	0.35	0.40	0.45	0.50
$Learning\_rate = 0.020$	0.9659	0.9591	0.9409	0.9227	0.9114	0.9069	0.8909	0.8341	0.7114	0.5750
$Learning\_rate = 0.025$	0.9705	0.9591	0.9432	0.9182	0.9205	0.8909	0.8205	0.8318	0.7796	0.5750
$Learning\_rate = 0.030$	0.9455	0.9546	0.9591	0.9341	0.9273	0.9205	0.8841	0.7523	0.6727	0.5477

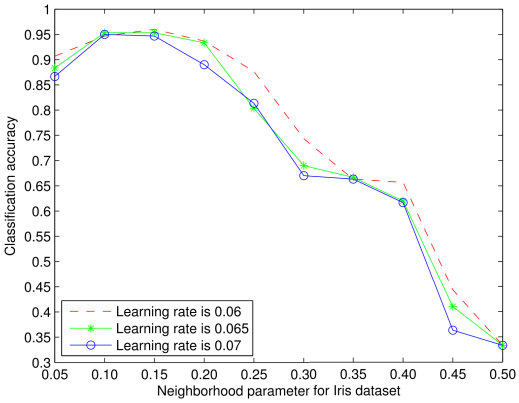


FIGURE 2. Influences of neighborhood parameter for Iris dataset.

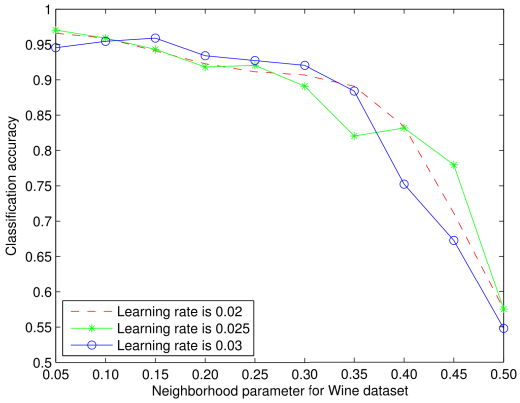


FIGURE 3. Influences of neighborhood parameter for Wine dataset.

of 0.065 is better than 0.07. When the neighborhood parameter is 0.15 and the learning rate is 0.06, the classification accuracy reaches the maximum value with 0.96. When the neighborhood parameters are 0.1-0.15 and the learning rate is 0.06, the classification effects of the granular convolutional classifier are better. And when the neighborhood parameter values are larger, the classification effects are worse.

It can be seen from Table 3 and Figure 3 that for Wine dataset, the classification effects are better while the neighborhood parameters are 0.05-0.15. When the neighborhood parameter is 0.05 and the learning rate is 0.025, the classification accuracy reaches the maximum value with 0.9705. When the neighborhood parameters are smaller, the classification effects of the granular convolutional classifier are better. And when the neighborhood parameters are larger, the classification effects are worse.

B. INFLUENCES OF LEARNING RATE

The granular convolutional classifier is related to the learning rate. If the learning rate is too large or too small, the training process is difficult to converge and affects the classification accuracy. In this subsection, the experiments test the impact of the learning rate on the classification accuracy. In these experiments, the learning rate of Iris dataset starts from 0.055 to 0.075 with an interval of 0.002. The learning rate of Wine dataset starts from 0.015 to 0.035 with an interval of 0.002. The neighborhood parameters are the better situations in the previous subsection. The Iris dataset uses neighborhood parameters with 0.1, 0.15, and 0.2, while the Wine dataset's neighborhood parameters are 0.05, 0.1, and 0.15. The experimental results are shown in Tables 4-5 and Figures 4-5, in which the horizontal axis is the learning rate and the vertical axis represents the classification accuracy.

As it can be seen from Table 4 and Figure 4, for Iris dataset, when the learning rates are varying from 0.057 to 0.065, their classification accuracy is slightly better than that of the learning rates varying from 0.071 to 0.075. It is illustrated from Table 5 and Figure 5, for Wine dataset, the classification effects are better when the learning rates are between 0.027 and 0.035; the classification effects are worse when the learning rates are between 0.015 and 0.023. These indicates that the granular convolutional classifier should tune a suitable learning rate for the specific dataset. Therefore, the values of learning rate play a key role in the learning process of the granular convolutional classifier.

C. CONVERGENCE ANALYSIS

The training of a granular convolutional classifier is a process of continuous iterative convergence. The experiments in this subsection mainly analyze the convergence of a training of a granular convolutional classifier. The neighborhood parameters and learning rates are the best situations in the previous subsection. The neighborhood parameter of Iris dataset

TABLE 4. Influences of learning rate for Iris dataset.

$Learning\_rate$	0.055	0.057	0.059	0.061	0.063	0.065	0.067	0.069	0.071	0.073	0.075
$Neighborhood\_parameter = 0.10$	0.9333	0.9600	0.9367	0.9533	0.9367	0.9500	0.9500	0.9633	0.9400	0.9333	0.9500
$Neighborhood\_parameter = 0.15$	0.9333	0.9433	0.9567	0.9567	0.9500	0.9467	0.9467	0.9600	0.9500	0.9433	0.9567
$Neighborhood\_parameter = 0.20$	0.9367	0.9500	0.9233	0.9467	0.9400	0.9300	0.9233	0.9167	0.9400	0.9300	0.9100

TABLE 5. Influences of learning rate for Wine dataset.

$Learning\_rate$	0.015	0.017	0.019	0.021	0.023	0.025	0.027	0.029	0.031	0.033	0.035
$Neighborhood\_parameter = 0.05$	0.7977	0.8886	0.9364	0.9432	0.9523	0.9591	0.9636	0.9546	0.9682	0.9455	0.9341
$Neighborhood\_parameter = 0.10$	0.8750	0.8932	0.9432	0.9364	0.9341	0.9477	0.9682	0.9614	0.9591	0.9477	0.9659
$Neighborhood\_parameter = 0.15$	0.8546	0.9159	0.9273	0.9318	0.9386	0.9477	0.9682	0.9455	0.9591	0.9455	0.9318

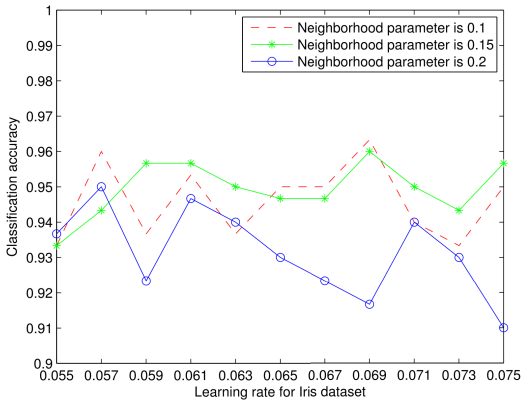


FIGURE 4. Influences of learning rate for Iris dataset.

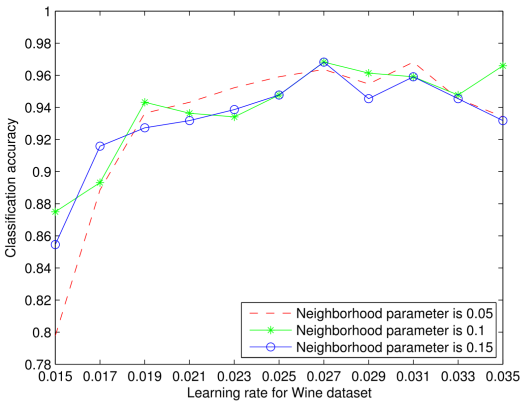


FIGURE 5. Influences of learning rate for Wine dataset.

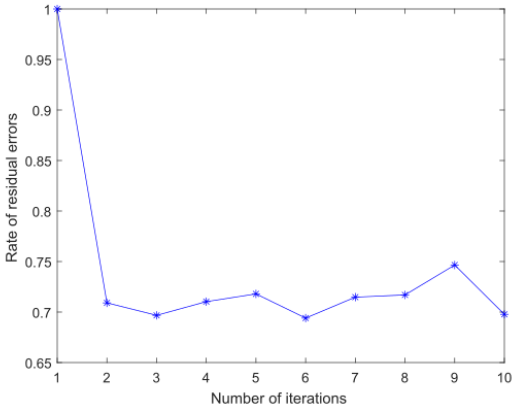


FIGURE 6. Convergence for Iris dataset.

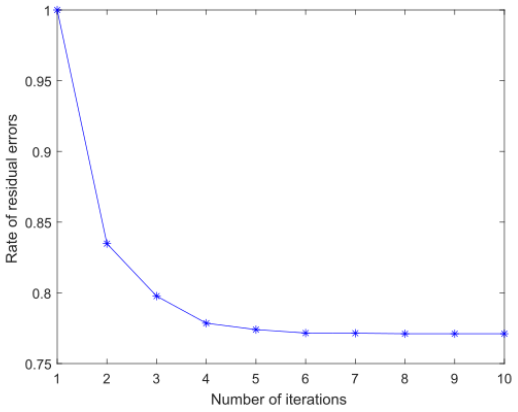


FIGURE 7. Convergence for Wine dataset.

is 0.15, while the learning rate is 0.065. The neighborhood parameter of Wine dataset is 0.05, and the learning rate is 0.025. The convergence index is a ratio of residual errors with formula e/e_{max}, where e is the current residual error, and e_{max} is the maximum residual error. The experimental results are shown in Figures 6 and 7. The abscissa is a number of iterations, and the ordinate represents a ratio of residual errors.

It can be seen from Figures 6 and 7 that the learning process of the granular convolutional classifier converges fast. For Iris dataset, its residual error converges in the second iteration. After that, it oscillates to a certain range. As for Wine dataset, its residual error converges to a fixed value in its eighth iteration.

D. ANALYSIS OF CLASSIFICATION ACCURACY

In order to test the classification accuracy of our proposed method, we compared the Granular Convolutional Classifier (GCC) with traditional KNN [41] and SVM [42] classifiers. We perform a five-cross validation method for testing these classifiers. The experimental data is randomly divided into five portions, four of which are used for training, one for testing. Exchanging another portion for testing, the remaining four portions are used for training. There are

total five tests, and each test result is the mean classification accuracy of testing samples. The results are shown in Figures 8 and 9. The ordinate indicates the classification accuracy and the abscissa presents the number of tests.

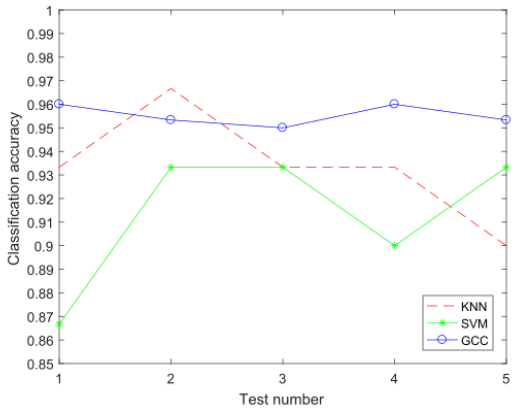


FIGURE 8. Classification for Iris dataset.

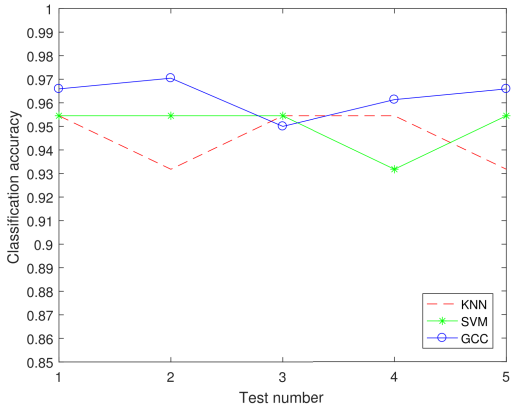


FIGURE 9. Classification for Wine dataset.

For Iris dataset from Figure 8, it can be seen that the classification accuracy of GCC is better than that of SVM. It is better than SVM algorithm in most cases. For Wine dataset from Figure 9, it also can be seen that the classification accuracy of the GCC is better than those of KNN and SVM in most cases.

VII. CONCLUSION

The traditional classifier is a numerical calculation, which does not involve operations of sets. Starting from the neighborhood granulation of samples, we proposed a new granular convolutional classifier by some set operations. Firstly, the neighborhood rough set mode is introduced, and the neighborhood granules and granule vectors are constructed in a classification system. The measures of granules and operations of granule vectors are proposed. Furthermore, the convolutional operation of the granule vector is defined for extracting features, and the positive and negative residual errors are achieved for a back propagation. We proposed a classifier by designing some back propagation rules to revise values of weights of granules. Finally, experiments are carried out from the aspects of neighborhood parameter, learning rate, convergence and classification accuracy. The results show that the newly proposed granular convolutional classifier can successfully classify samples and obtain a better classification performance under the conditions of suitable granulation parameters and learning rates. In the future work, the mode of neural network will be introduced to develop new parameter adjustment methods, and new back propagation rules are studied for the constructions of granular neural network classifiers. It is also possible to study the local granulation method for constructing local granule vectors, and apply the classification method proposed in this paper to the big data systems.

REFERENCES

[1] G. Forney, "Convolutional codes I: Algebraic structure," *IEEE Trans. Inf. Theory*, vol. IT-16, no. 6, pp. 720–738, Nov. 1970.
[2] T. Xie and J. C. Grossman, "Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties," *Phys. Rev. Lett.*, vol. 120, no. 14, pp. 1–6, 2018.
[3] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
[4] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 1–20, 2016.
[5] M. Thom and F. Gritschneder, "Rapid exact signal scanning with deep convolutional neural networks," *IEEE Trans. Signal Process.*, vol. 65, no. 5, pp. 1235–1250, Mar. 2017.
[6] V. Bhargava, "Forward error correction schemes for digital communications," *IEEE Commun. Mag.*, vol. 21, no. 1, pp. 11–19, Jan. 1983.
[7] Y. S. Vang and X. Xie, "HLA class I binding prediction via convolutional neural networks," *Bioinformatics*, vol. 33, no. 17, pp. 2658–2665, 2017.
[8] N. Tajbakhsh, J.Y. Shin, S. R. Gurudu, R. T. Hurst, C. B. Kendall, M. B. Gotway, and J. Liang, "Convolutional neural networks for medical image analysis: Full training or fine tuning?" *IEEE Trans. Med. Imag.*, vol. 35, no. 5, pp. 1299–1312, May 2016.
[9] H. Zhang and V. M. Patel, "Convolutional sparse and low-rank coding-based image decomposition," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2121–2133, May 2018.
[10] E. Basaeed, H. Bhaskar, and M. Al-Mualla, "Supervised remote sensing image segmentation using boosted convolutional neural networks," *Knowl.-Based Syst.*, vol. 99, pp. 19–27, May 2016.
[11] Y. Wang, W. Ren, and H. B. Wang, "Anisotropic second and fourth order diffusion models based on convolutional virtual electric field for image denoising," *Comput. Math. With Appl.*, vol. 66, no. 10, pp. 1729–1742, 2013.
[12] Y. Cao, Y. Chen, and D. Khosla, "Spiking deep convolutional neural networks for energy-efficient object recognition," *Int. J. Comput. Vis.*, vol. 113, no. 1, pp. 54–66, 2015.
[13] S.-C. B. Lo, H.-P. Chan, J.-S. Lin, H. Li, M. T. Freedman, and S. K. Mun, "Artificial convolution neural network for medical image pattern recognition," *Neural Netw.*, vol. 8, nos. 7–8, pp. 1201–1214, 1995.
[14] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Trans. Neural Netw.*, vol. 8, no. 1, pp. 98–113, Jan. 1997.
[15] Y. Zhang, K. Shang, J. Wang, N. Li, and M. M. Y. Zhang, "Patch strategy for deep face recognition," *IET Image Process.*, vol. 12, no. 5, pp. 819–825, 2018.
[16] X. Gao and T. Zhang, "Unsupervised learning to detect loops using deep neural networks for visual SLAM system," *Auton. Robots*, vol. 41, no. 1, pp. 1–18, 2017.
[17] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 1725–1732.

[18] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.

[19] F. Commandeur, M. Goeller, J. Betancur, S. Cadet, M. Doris, X. Chen, D. S. Berman, P. J. Slomka, B. K. Tamarappoo, and D. Dey, "Deep learning for quantification of epicardial and thoracic adipose tissue from non-contrast CT," *IEEE Trans. Med. Imag.*, vol. 37, no. 8, pp. 1835–1846, Aug. 2018.

[20] L. Wen, X. Li, L. Gao, and Y. Zhang, "A new convolutional neural network-based data-driven fault diagnosis method," *IEEE Trans. Ind. Electron.*, vol. 65, no. 7, pp. 5990–5998, Jul. 2018.

[21] M. D. Ferreira, D. C. Corrêa, L. G. Nonato, and, R. F. de Mello, "Designing architectures of convolutional neural networks to solve practical problems," *Expert Syst. Appl.*, vol. 94, pp. 205–217, Mar. 2018.

[22] M. Kabkab, E. Hand, and R. Chellappa, "On the size of convolutional neural networks and generalization performance," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 3572–3577.

[23] Z. Pawlak, "Rough sets," *Int. J. Comput. Inf. Sci.*, vol. 11, no. 5, pp. 341–356, Oct. 1982.

[24] Z. Pawlak and A. Skowron, "Rough sets and Boolean reasoning," *Inf. Sci.*, vol. 177, pp. 41–73, Jan. 2007.

[25] 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.

[26] T. Y. Lin, "Data mining: Granular computing approach," in *Methodologies for Knowledge Discovery and Data Mining* (Lecture Notes in Computer Science), vol. 1574, no. 1. Berlin, Germany: Springer, 1999, pp. 24–33.

[27] T. Y. Lin and L. A. Zadeh, "Special issue on granular computing," *IEEE Trans. Fuzzy Syst.*, vol. 14, no. 2, p. 360, Aug. 2006.

[28] Y. Y. Yao, "Information granulation and rough set approximation," *Int. J. Intell. Syst.*, vol. 16, no. 1, pp. 87–104, 2001.

[29] Y. Y. Yao, N. Zhang, D. Q. Miao, and F. F. Xu, "Set-theoretic approaches to granular computing," *Fundamenta Informaticae*, vol. 115, nos. 2–3, pp. 247–264, 2012.

[30] W. Pedrycz, B. J. Park, and S. K. Oh, "The design of granular classifiers: A study in the synergy of interval calculus and fuzzy sets in pattern recognition," *Pattern Recognit.*, vol. 41, no. 12, pp. 3720–3735, 2008.

[31] S. B. Roh, W. Pedrycz, and T.-C. Ahn, "A design of granular fuzzy classifier," *Expert Syst. Appl.*, vol. 41, no. 15, pp. 6786–6795, 2014.

[32] H. Fujita, A. Gaeta, V. Loia, and F. Orciuoli, "Resilience analysis of critical infrastructures: A cognitive approach based on granular computing," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1835–1848, May 2019.

[33] J. Liang, J. Wang, and Y. Qian, "A new measure of uncertainty based on knowledge granulation for rough sets," *Inf. Sci.*, vol. 179, no. 4, pp. 458–470, 2009.

[34] 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.

[35] Q. H. Hu, D. Yu, and Z. Xie, "Neighborhood classifiers," *Expert Syst. With Appl.*, vol. 34, no. 2, pp. 866–876, 2008.

[36] 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, 2008.

[37] P. Zhu and Q. H. Hu, "Adaptive neighborhood granularity selection and combination based on margin distribution optimization," *Inf. Sci.*, vol. 249, pp. 1–12, Nov. 2013.

[38] Y. Chen, N. Qin, W. Li, and F Xu, "Granule structures, distances and measures in neighborhood systems," *Knowl.-Based Syst.*, vol. 165, pp. 268–281, Feb. 2018.

[39] Y. M. Chen, Q. X. Zhu, K. S. Wu, and S. Z. Zhu, "A binary granule representation for uncertainty measures in rough set theory," *J. Intell. Fuzzy Syst.*, vol. 28, no. 2, pp. 867–878, 2015.

[40] Y. M. Chen, Y. Zhuang, S. Z. Zhu, W. Li, and C. H. Tang, "A granulated fuzzy rough set and its measures," *J. Intell. Fuzzy Syst.*, vol. 35, no. 5, pp. 5667–5677, 2018.

[41] W.-J. Hwang, "Fast kNN classification algorithm based on partial distance search," *Electron. Lett.*, vol. 34, no. 21, pp. 2062–2063, Oct. 1998.

[42] F. Friedrichs and C. Igel, "Evolutionary tuning of multiple SVM parameters," *Neurocomputing*, vol. 64, pp. 107–117, Mar. 2005.

[43] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[44] O. A. Alzubi, J. A. Alzubi, S. Tedmori, H. Rashaideh, and O. Almomani, "Consensus-based combining method for classifier ensembles," *Int. Arab J. Inf. Technol.*, vol. 15, no. 1, pp. 76–86, Jan. 2018.

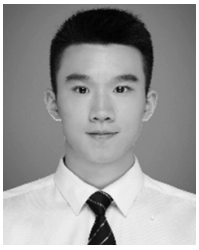
[45] R. Wang, N. Xiu, and C. Zhang, "Greedy projected gradient-Newton method for sparse logistic regression," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published.

[46] J. A. Alzubi, "Diversity-based boosting algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 5, pp. 524–529, 2016.

[47] J. A. Alzubi, "Diversity based improved bagging algorithm," in *Proc. Int. Conf. Eng. MIS*, Istanbul, Turkey, 2015, p. 35.

[48] Y. Liu, W. L. Huang, Y. L. Jiang, and Z. Y. Zeng, "Quick attribute reduct algorithm for neighborhood rough set model," *Inf. Sci.*, vol. 271, pp. 65–81, Jul. 2014.

YUMIN CHEN received the M.E. degree in computer application technology from Nanchang University, China, in 2005, and the Ph.D. degree in pattern recognition and intelligence system from Tongji University, China, in 2010. From November 2018 to November 2019, he was a Visiting Scholar with the Department of Mathematics and Statistics, Mississippi State University, USA. He was supported by the Education Department of Fujian Province. He is currently a Professor with the College of Computer and Information Engineering, Xiamen University of Technology. He has published over 50 articles in refereed journals and conferences. His research interests include machine learning, rough sets, and granular computing.



XIAO ZHANG received the B.S. degree in information and computing science from Shanghai Maritime University, China, in 2018. He is currently pursuing the M.E. degree with the College of Computer and Information Engineering, Xiamen University of Technology. His research interests include machine learning, image processing, and pattern recognition.

WEI LI received the Ph.D. degree in basic theory of artificial intelligence from Xiamen University, China, in 2013. He was also a Visiting Scholar with the Department of Computer Science, University of Massachusetts Boston, USA, from June 2018 to June 2019. He is an Associate Professor and a Master Supervisor with the College of Computer and Information Engineering, Xiamen University of Technology, Xiamen, China. His research interests include artificial intelligence, computer graphics, mobile security and privacy, and granular computing. He is also a member of the China Computer Federation (CCF) and the Chinese Association for Artificial Intelligence (CAAI).

SHUNZHI ZHU received the Ph.D. degree in information systems from Xiamen University, China, in 2007. He was a Visiting Scholar with Florida International University, USA. He is currently a Professor with the College of Computer and Information Engineering, Xiamen University of Technology. His research interests include data mining, information recommendation, GIS applications, and system engineering.