IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 23, 2017, accepted August 9, 2017, date of publication August 15, 2017, date of current version September 6, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2739807

Sparse Coefficient-Based *k*-Nearest Neighbor Classification

HONGXING MA[1,2], JIANPING GOU[3], (Member, IEEE), XILI WANG[1], JIA KE[3], AND SHAONING ZENG[4]

[1]College of Computer Science, Shaanxi Normal University, Xi'an 710119, China
[2]College of Electrical and Information Engineering, North Minzu University, Yinchuan 750021, China
[3]School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang 212013, China
[4]School of Information Science and Technology, Huizhou University, Huizhou 516007, China

Corresponding author: Jianping Gou (goujianping@ujs.edu.cn)

This work was supported in part by National Science Foundation of China under Grant 61502208 and Grant 41471280, in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20150522, in part by the North Minzu University School Project under Grant 2016DX008, in part by the China Postdoctoral Science Foundation under Grant 2015M570411, in part by the Research Foundation for Talented Scholars of Jiangsu University under Grant 14JDG037, and in part by the Open Foundation of Artificial Intelligence Key Laboratory of Sichuan Province under Grant 2017RYJ04.

ABSTRACT *K*-nearest neighbor rule (KNN) and sparse representation (SR) are widely used algorithms in pattern classification. In this paper, we propose two new nearest neighbor classification methods, in which the novel weighted voting methods are developed for making classification decisions on the basis of sparse coefficients in the SR. Since the sparse coefficients can well reflect the neighborhood structure of data, we mainly utilize them to design classifier in the proposed methods. One proposed method is called the coefficient-weighted KNN classifier, which adopts sparse coefficients to choose KNNs of a query sample and then uses the coefficients corresponding to the chosen neighbors as their weights for classification. Another new method is the residual-weighted KNN classifier (RWKNN). In the RWKNN, KNNs of a query sample are first determined by sparse coefficients, and then, we design a novel residual-based weighted voting method for the KNN classification. The extensive experiments are carried out on many UCI and KEEL data sets, and the experimental results show that the proposed methods perform well.

INDEX TERMS K-nearest neighbor rule, sparse coefficient, weighted voting, pattern classification.

I. INTRODUCTION

K-nearest neighbor rule is one of the top ten algorithms in data mining [1]. Because of the simplicity and effectiveness of KNN, it has been widely used in pattern recognition since it was first proposed in [2]. Also, it has asymptotically optimal performance in the Bayes sense for good classification [2], [3]. Although KNN has such superiorities, it still has two key issues that are the sensitiveness to the neighborhood size k and the simple majority vote in the k-neighborhood region for the classification decision [1], [4]–[6].

In general, the sensitiveness to k can always be produced by the different neighborhood selection criteria, the simple similarity measure by Euclidean distance and outliers especially in the small sample size cases. The general neighborhood selection only considers the similarities among samples. Using such neighborhood selection, the performance of the KNN-based classifiers is very sensitive to the neighborhood size k. As an alternative neighborhood, nearest centroid neighborhood (NCN) is another effective way of selecting k nearest neighbors [7]–[10]. NCN considers both the proximity and geometrical distribution of k nearest neighbors of the query samples simultaneously. Such k-nearest centroid neighbor classifiers can perform well and are robust to k. To address the issue with Euclidean distance similarity measure, distance metric learning-based nearest neighbor classifiers were proposed with good classification performance [11], [12]. Since the classification decisions of KNN-based methods are often determined by the simple majority vote in the local region of neighborhood, the performance can be easily degraded and very sensitive to k due to the existing outliers. For the simple majority vote, k neighbors are equally served for making the classification decision. As we know, the closer neighbors are generally more similar and should be given more weight for classification. To improve the simple majority vote, the distance-weighted voting methods for KNN were proposed in [4] and [13].

Furthermore, in the small sample size cases with the existing outliers and imbalance data, the performance of KNN-based nonparametric classifiers is severely damaged [14] and the sensitiveness to the neighborhood size k is very obvious. In regard to this problem, a local mean-based k-nearest neighbor method (LMKNN) was proposed to achieve the good classification with robustness to outliers [15]. Its classification decision is realized by utilizing the local mean vector of k nearest neighbors from each class. Using the idea of LMKNN, another new LMKNN method based on the NCN was successfully introduced with promising classification performance in [10]. By calculating the weighted sum of distances of k nearest neighbors per class, a pseudo nearest neighbor method (PNN) was designed in [16]. Based on PNN and LMKNN, a local mean-based pseudo nearest neighbor method (LMPNN) was proposed in [5]. LMPNN obtains the very satisfactory classification in many pattern recognition problems, such as in [17]. Recently, to aim at the outliers and the choice of k, a new k-harmonic nearest neighbor classifier based on the multi-local mean vectors was proposed by using the idea of LMPNN and the harmonic mean distance in [18]. Besides, when the samples from different the classes are nonseparable or overlapping, the classification decision in the neighborhood region could be wrong. To address the issue, a new KNN-based method was developed by calculating local categorical probability centers of each query sample in [19] and then the class of the query sample was determined by employing the distances from the query sample to these centers.

Up to now, sparse representation [20]–[22] has been well successfully applied in pattern recognition such as in face recognition [23]–[26]. One representative SR-based classification algorithm is the sparse representation-based classification method (SRC) [23]. Due to the advantage of sparsity for classification, there exist some SR-based nearest neighbor classification methods [27], [28]. It has been argued that the sparse representation coefficients as the good similarities among samples can reflect the neighborhood structure of data and have more discrimination information [27], [29], [30]. If two samples are similar, the corresponding sparse coefficients should also be close to each other [30]. In SR-based classification, each query sample is sparsely represented by all the training samples in general. If a query sample is very similar with one training sample, the training sample will have a large sparse coefficient to represent the query sample. Thus, the coefficients have been well employed for making classification decision [27], [31], [32]. To match well with SRC, sum of coefficients (SoC), as a classification decision was proposed by fully taking advantage of sparse representation coefficients in [31] and it performs well. Moreover, since the sparse reconstructive relationship of data through sparse representation contains natural geometrical similarity structure of data, SR-based graph construction has been well used in graph embedding [33]–[36]. In addition to sparse representation, collaborative representation (CR) [37] is another representation method for classification [38], [39]. Due to good properties of collaborative representation in pattern recognition, there are two CR-based KNN classification methods [40], [41]. The one is the coarse to fine k nearest neighbor classifier that uses two-phase representation to find nearest neighbors [40]. The other one is the collaborative representation-based nearest neighbor classifier (CRNN) [41], that mainly uses representation coefficients to find k nearest neighbors of a query sample.

With regard to the problems about the neighborhood selection and the simple majority vote in the KNN-based classification, two novel k-nearest neighbor methods motivated by the good advantages of sparse representation are proposed, with the aim of improving the classification performance, especially in the small sample size cases. The two proposed methods are called the coefficient-weighted k-nearest neighbor classifier (CWKNN) and the residual-weighted k-nearest neighbor classifier (RWKNN), respectively. In the proposed methods, since the sparse coefficients in SR can reflect the similarities among samples, the sparse coefficients of the sparse representation of each query sample are employed to choose k nearest neighbors of each query sample. The chosen neighbors are representative to represent each query sample for the useful classification. In the CWKNN, the sparse coefficients are viewed as the weights of the corresponding neighbors, and then the coefficient-weighted voting is used for the classification decision. In the RWKNN, we calculate the reconstructive residual distances between each query sample and the k-nearest neighbors and then design the residual distance-weighted function to compute the weight of each neighbor. The residual-weighted voting combining the sparse coefficients is adopted for the classification decision in the RWKNN. The extensive experiments on UCI and KEEL data sets of the competing methods are conducted to verify the classification performance of the proposed methods. Experimental results demonstrate the proposed methods perform very well.

The rest of this article is organized as follows. Section II presents the related works. Section III describes the proposed methods. Section IV further analyzes the proposed methods. Sections V and VI provide the extensive experimental results and some discussions, respectively. Finally, the conclusion is given in Section VII.

II. RELATED METHODS

In this section, we briefly review the related methods including KNN, sparse representation-based classification (SRC) and two representation-based nearest neighbor classifiers (*i.e.*, LRMNN and CRNN). For description convenience, we first introduce some notations. Let $T = \{(x_i, l_i)\}_{i=1}^{n}$ denote the training set with n training samples in M classes $C = \{c_1, c_2, \cdots, c_M\}$, where $x_i \in R^d$ is a training sample with the corresponding class label $l_i \in C$ in the d-dimensional feature space. The training set from the j-th class is indicated as $T_j = \{(x_i^j, l_i^j)\}_{i=1}^{n_j}$ and n_j is the number of training samples from j-th class.

A. KNN

KNN is a very simple and effective nonparametric classification algorithm in many pattern recognition problems. In KNN, the class label of the query sample is generally decided by the simple majority vote of k nearest neighbors chosen from the training set.

Given a query sample $y \in R^d$, k nearest neighbors of y are first searched from all the training samples by computing the Euclidean distance between y and x_i

$$d(y, x_i) = \|y - x_i\|_2. \tag{1}$$

The training samples corresponding to the first k smallest distances are chosen as k nearest neighbors of the query sample, denoted as $T_k = \{(x_i^{NN}, l_i^{NN})\}_{i=1}^k$. Then the class label l_y of y is predicted by the majority vote of k nearest neighbors as

$$l_y = \arg\max_c \sum_{(x_i^{NN},\, l_i^{NN}) \in T_k} \delta(c = l_i^{NN}). \tag{2}$$

where $c \in C$ and $\delta(c = l_i^{NN})$ is the Dirac delta function that takes a value of one if $c = l_i^{NN}$ and zero otherwise, and l_i^{NN} is the class label of the i-th nearest neighbor.

As an improvement, the distance-weighted k-nearest neighbor rule (WKNN) was first proposed by a distance-weighted voting method [13]. The distance-weighted function is defined as:

$$w_i = \begin{cases} \dfrac{d(y, x_k^{NN}) - d(y, x_i^{NN})}{d(y, x_k^{NN}) - d(y, x_1^{NN})}, & if\ d(y, x_k^{NN}) \neq d(y, x_1^{NN}), \\ 1, & if\ d(y, x_k^{NN}) = d(y, x_1^{NN}), \end{cases} \tag{3}$$

where w_i is the distance-based weight of x_i^{NN} as the contribution to classifying y. It is clear that the range of the weight w_i is from one for the first neighbor to zero for k-th neighbor, and one neighbor with smaller distance has greater weight. Then, the query y is assigned into the dominated class that has the largest sum of weights among k nearest neighbors as follows:

$$l_y = \arg\max_c \sum_{(x_i^{NN},\, l_i^{NN}) \in T_k} w_i \times \delta(c = l_i^{NN}). \tag{4}$$

B. SRC

Sparse representation-based classification (SRC) is very famous and powerful in the latest classification methods [23]. In SRC, a query sample y is first represented by a linear sparse combination of all the training samples $\{x_1, x_2, \ldots, x_n\}$ and then classified by the sparse reconstruction residual error between each class and the query sample. The query sample y is sparsely represented as

$$y = s_1 x_1 + s_2 x_2 +, \ldots, +s_n x_n, \tag{5}$$

where s_i is the sparse representation coefficient associated with the training sample x_i to represent y and $i \in 1, 2, \ldots, n$. In general, the sparse coefficient vector $s = [s_1, s_2, \ldots, s_n]^T$ is optimized with L_1-norm constraint as in [20], [21]:

$$\begin{cases} \tilde{s} = \arg\min_s \|s\|_1 \\ s.t.\ y = Xs, \end{cases} \tag{6}$$

where $X = [x_1, x_2, \ldots, x_n]$ is the matrix of all the training samples.

In terms of the optimal $\tilde{s}$, y is classified by the minimum class-specific residual error as [23]:

$$l_y = \arg\min_{c_i} \|y - X\sigma^i(\tilde{s})\|_2, \tag{7}$$

where $\sigma^i(\tilde{s})$ is a new vector that has nonzero entries the same as the ones in $\tilde{s}$ corresponding to class c_i. To get the more discrimination from sparse coefficients, the decision rule called sum of coefficients (SoC) was proposed as [31]:

$$l_y = \arg\max_{c_i} \sum \sigma^i(\tilde{s}). \tag{8}$$

where $\sum \sigma^i(\tilde{s})$ is the sum of coefficients of the training samples from class c_i. The query sample y is assigned into the class with the largest sum of coefficients.

C. LRMNN

Linear reconstruction measure steered nearest neighbor classification (LRMNN) [27] is an extension of the KNN-based classification. Since linear reconstruction measure (LRM) can provide more meaningful information than the conventional similarity measures such as Euclidean distance, the linear reconstruction coefficients (*i.e.* the representation coefficients) in LRM are employed to determine the nearest neighbors of each query sample for classification.

The general LRM model of a query sample y is uniformly formulated with regularized terms as follows:

$$\min \left\{ \|y - Xa\|_2^2 + \gamma \|a\|_p \right\}, \tag{9}$$

where the vector of the linear reconstruction coefficients of y is $a = [a_1, a_2, \ldots, a_n]^T$ and a_i is the representation coefficient associated with training sample x_i, γ is the regularization parameter, and $p = 1$ or 2 in general. After solving Eq. (9), the optimal vector of a is $\bar{a} = [\bar{a}_1, \bar{a}_2, \ldots, \bar{a}_n]^T$. Then, the classification decision rule of LRMNN is obtained as

$$l_y = \arg\max_{l_i} \|\bar{a}_i\|. \tag{10}$$

The query sample y is classified into the class of the training sample x_i that has largest $\|\bar{a}_i\|$, and x_i is the most representative nearest neighbor of y among all training samples.

D. CRNN

Collaborative representation-based nearest neighbor classifier (CRNN) on the basis of collaborative representation [37] is proposed for hyperspectral image classification [41].

In CRNN, the weighted linear collaborative combination of a query sample y is formulated as

$$\bar{b} = \arg\min_{b} \{\|y - Xb\|_2^2 + \tau \|W_y b\|_2^2\}, \tag{11}$$

where $b = [b_1, b_2, \ldots, b_n]^T$ is the vector of the weighted linear collaborative representation coefficients and b_i is the representation coefficient associated with training sample x_i, τ is the regularization parameter, and W_y is a biasing Tikhonov matrix. The biasing Tikhonov matrix is defined as

$$W_y = \begin{bmatrix} \|y - x_1\|_2 & & 0 \\ & \ddots & \\ 0 & & \|y - x_n\|_2 \end{bmatrix}. \tag{12}$$

According to the optimal $\bar{b}$ of b by solving Eq. (11), k nearest neighbors of y corresponding to the k largest coefficients in $\bar{b}$ are found. Finally, the class label of the query sample y is determined by the majority vote of k nearest neighbors using Eq. (2).

III. THE PROPOSED METHODS

In this section, we first present the motivation of the proposed CWKNN and RWKNN methods, and then describe them in details.

A. THE MOTIVATION

The proposed CWKNN and RWKNN methods on basis of the sparse representation coefficients for the testing samples are mainly inspired by the superiorities of sparse representation in pattern classification. To clearly elaborate why the proposed methods are meaningfully designed, we analyze the advantages of sparse representation for the proposed classification methods in this subsection.

First of all, the SRC and its extensions have shown that sparse representation has more power of natural pattern discrimination for good pattern classification [23]–[26], [38] and the discrimination information of data is mainly preserved in the sparse coefficients [33]. Using this advantage of sparse representation to capture more discrimination information of data for the query samples, each query sample is first represented by the linear sparse combination of all the training samples in the proposed CWKNN and RWKNN for classification.

Secondly, sparse representation obtains the sparse reconstructive relationship of the data that can reflect intrinsic geometric similarity information from the data and is useful for classification [33]. Due to the geometric property of data by sparse representation, the graph construction methods with sparsity have been well developed in graph embedding-based dimensionality reduction [33]–[36]. In fact, the weights of edges between any pairs of the sample points in these sparse representation-based graphs are measured by the corresponding sparse coefficients, and sparse coefficients can well reflect the neighborhood structure of samples that is hidden in data. Furthermore, the sparse coefficients as a good similarity measure of data instead of the conventional similarity measures such as Euclidean distance have been successfully proven in [27] and [29], and the sparse coefficients instead of residuals are well adopted to make classification decisions in the representation-based classification [27], [28], [31], [32], [39]. Besides, it has been argued that the similar samples should also have similar sparse coefficients [30]. Due to these facts that the sparse coefficients can well reflect the similarities among samples and contain natural discrimination information, the sparse coefficients of sparse representation of each query sample using all training samples are fully employed to seek its k nearest neighbors and further used for classifier design in the proposed CWKNN and RWKNN methods.

B. THE PROPOSED CWKNN

In the CWKNN, each query sample y is first sparsely represented by using all the training samples as

$$y = X\beta, \tag{13}$$

where $\beta = [\beta_1, \beta_2, \ldots, \beta_n]^T$ is the sparse coefficient vector including the sparse coefficients of all the training samples to represent y. The coefficient β_i of the sample x_i is regarded as the contribution to representing y. If x_i is more similar to y, the contribution β_i could be larger. Indeed, it has been argued that the sparse coefficients can well reflect the similarities among the samples [27], [29], [30]. The sparse representation of y is solved with L_1-norm regularization as follows:

$$\bar{\beta} = \arg\min_{\beta} \{\|y - X\beta\|_2^2 + \lambda \|\beta\|_1\}, \tag{14}$$

where $\bar{\beta} = [\bar{\beta}_1, \bar{\beta}_2, \ldots, \bar{\beta}_n]^T$. It should be noted that Eq. (14) is essentially the Lasso objective function with L_1-norm regularization [42], which has been used in [27] and [43].

After obtaining the sparse coefficients in Eq. (14), they are utilized to choose the k nearest neighbors of each query sample. The top k largest coefficients from $\bar{\beta}$ are chosen and denoted as the set $\tilde{\beta} = \{\tilde{\beta}_1, \tilde{\beta}_2, \ldots, \tilde{\beta}_k\}$, and the corresponding training samples as k nearest neighbors of y are also denoted as the set $T_k = \{(x_i^{NN}, l_i^{NN})\}_{i=1}^k$. Note that the chosen $\tilde{\beta}_i$ is always equal or greater than zero. Meanwhile, the contribution $\tilde{\beta}_i$ of the i-th nearest neighbor x_i^{NN} to representing the query sample y is naturally treated as the weight of x_i^{NN} to y.

Finally, the classification decision of the CWKNN is defined as

$$l_y = \arg\max_{c} \sum_{(x_i^{NN},\, l_i^{NN}) \in T_k} \tilde{\beta}_i \times \delta(c = l_i^{NN}), \tag{15}$$

where l_y is the class label of y. The query sample y is classified into the class that has the largest sum of weights among the classes that k nearest neighbors belong to.

As stated above, the CWKNN method adopts the sparse coefficients to seek nearest neighbors and employs sparse coefficient-weighted voting for classification. Note that although the CWKNN seems to be similar with the LRMNN

with L_1-norm regularization [27], they are different in the aspects of determining nearest neighbors and making classification decision that are detailedly stated in section VI-A. The proposed CWKNN can be summarized by pseudo codes in Algorithm 1.

Algorithm 1 The Algorithm of the Coefficient-Weighted k-Nearest Neighbor Classifier

Require:

The training set $T = \{(x_i, l_i)\}_{i=1}^{n}$ where $l_i \in C$, the set of M class labels $C = \{c_1, c_2, \cdots, c_M\}$, a query sample y, the neighborhood size k.

Ensure:

Predict the class label of a query sample by sparse coefficient-weighted voting.

Step 1: Solve the sparse representation of y using all the training samples.

$$\bar{\beta} = \arg\min_{\beta} \left\{ \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\}$$

Step 2: Search k nearest neighbors of y using the sparse coefficients $\bar{\beta} = [\bar{\beta}_1, \bar{\beta}_2, \ldots, \bar{\beta}_n]^T$.

Step 2.1: Find first k largest sparse coefficients $\tilde{\beta} = \{\tilde{\beta}_1, \tilde{\beta}_2, \ldots, \tilde{\beta}_k\}$ from $\bar{\beta}$

Step 2.2: Choose the k nearest neighbors $T_k = \{(x_i^{NN}, l_i^{NN})\}_{i=1}^{k}$ corresponding to k largest sparse coefficients $\tilde{\beta}$.

Step 3: Classify y to the class c with the largest sum of weights among all the class.

$$l_y = \arg\max_{c} \sum_{(x_i^{NN},\, l_i^{NN}) \in T_k} \tilde{\beta}_i \times \delta(c = l_i^{NN})$$

C. THE PROPOSED RWKNN

In the RWKNN, k nearest neighbors of each query sample are first chosen by the same way as in the CWKNN through the sparse representation of each query sample using all the training samples. In terms of Eqs. (13) and (14), k nearest neighbors of the given query sample y are also indicated as the set $T_k = \{(x_i^{NN}, l_i^{NN})\}_{i=1}^{k}$ and the corresponding sparse coefficients are the set $\tilde{\beta} = \{\tilde{\beta}_1, \tilde{\beta}_2, \ldots, \tilde{\beta}_k\}$.

Using the k nearest neighbors and the corresponding sparse coefficients of the query sample y, the reconstructive residual distances between nearest neighbors and y are defined as

$$d_r(y, x_i^{NN}) = \|y - \tilde{\beta}_i x_i^{NN}\|_2. \tag{16}$$

In fact, the reconstructive residual distance $d_r(y, x_i^{NN})$ has been regarded as the contribution to representing the query sample y [38]. Let the obtained k reconstructive residual distances corresponding to k nearest neighbors be the set $\{d_r(y, x_1^{NN}), d_r(y, x_2^{NN}), \ldots, d_r(y, x_k^{NN})\}$. Then, among the k reconstructive residual distances, the largest and smallest ones are found as

$$d_{max}^{NN} = \max \left\{ d_r(y, x_i^{NN}) \right\}, \quad i = 1, 2, \ldots, k, \tag{17}$$

and

$$d_{min}^{NN} = \min \left\{ d_r(y, x_i^{NN}) \right\}, \quad i = 1, 2, \ldots, k. \tag{18}$$

The residual distance-weighted function is defined as follows:

$$wr_i = \begin{cases} \dfrac{d_{max}^{NN} - d_r(y, x_i^{NN})}{d_{max}^{NN} - d_{min}^{NN}}, & \text{if } d_{max}^{NN} \neq d_{min}^{NN}, \\ 1, & \text{if } d_{max}^{NN} = d_{min}^{NN}, \end{cases} \tag{19}$$

where wr_i is the weight of nearest neighbor x_i^{NN}. As can be seen in Eq. (19), the weights of nearest neighbors are scaled from 1 to 0. Furthermore, a neighbor with smaller reconstructive residual distance has larger weight than the one with greater reconstructive residual distance.

Using the sparse coefficients and the residual distance weights of k nearest neighbors, the classification decision of the RWKNN is defined as

$$l_y = \arg\max_{c} \sum_{(x_i^{NN},\, l_i^{NN}) \in T_k} wr_i \times \tilde{\beta}_i \times \delta(c = l_i^{NN}). \tag{20}$$

Finally, the class of the query y is the class with the largest sum of weights among the classes that k nearest neighbors belong to.

As argued above, the proposed RWKNN first uses sparse coefficients to choose k nearest neighbors of each query sample, and then designs the residual distance-weighted voting combining with sparse coefficients for classification. The proposed RWKNN can be summarized by pseudo codes in Algorithm 2.

IV. THE ANALYSIS OF THE PROPOSED METHODS

In this section, to emphasize the superior classification performance of the proposed CWKNN and RWKNN methods in pattern recognition, we analyze their rationale and advantages by comparing them with KNN and WKNN, and further show the representation coefficients instead of the conventional Euclidean distance as a good similarity measure.

From the point of view of sparse coefficients, the way of choosing the nearest neighbors of the query samples for classification in the proposed methods is reasonable and effective. As argued in [27] and [29], sparse coefficients can well reflect the neighborhood structure of data and represent the similarities among the samples. If two samples are similar, their corresponding sparse codes should also be very close to each other [30]. In terms of Eqs. (13) and (14), the given query sample y can be represented as $y = \sum_{i=1}^{n} \bar{\beta}_i x_i$, where $\bar{\beta}_i$ is optimized sparse coefficient of x_i to represent y. Note that y and x_i are normalized by L_2-norm. According to the theoretical analysis in [27] that the representation coefficients can availably indicate the true neighbors of each query sample, and x_i could not be the neighbor of y when $\bar{\beta}_i = 0$. However, in such case the sample x_i with very small similarity (*i.e.*, $\bar{\beta}_i = 0$) could be from the different or the same classes of y. If $\bar{\beta}_i \neq 0$, y can be further represented as

Algorithm 2 The Algorithm of the Residual-Weighted k-nearest Neighbor Classifier

Require:

The training set $T = \{(x_i, l_i)\}_{i=1}^{n}$ where $l_i \in C$, the set of M class labels $C = \{c_1, c_2, \cdots, c_M\}$, a query sample y, the neighborhood size k.

Ensure:

Predict the class label of a query sample by residual distance-weighted voting.

Step 1: Solve the sparse representation of y using all the training samples.

$$\bar{\beta} = \arg\min_{\beta} \{\|y - X\beta\|_2^2 + \lambda\|\beta\|_1\}$$

Step 2: Search k nearest neighbors of the query $T_k = \{(x_i^{NN}, l_i^{NN})\}_{i=1}^{k}$ corresponding to k largest sparse coefficients $\tilde{\beta} = \{\tilde{\beta}_1, \tilde{\beta}_2, \ldots, \tilde{\beta}_k\}$ from $\bar{\beta} = [\bar{\beta}_1, \bar{\beta}_2, \ldots, \bar{\beta}_n]^T$.

Step 3: Calculate the residual distances between k nearest neighbors and y.

for $j = 1$ to k **do**

$d_r(y, x_i^{NN}) = \|y - \tilde{\beta}_i x_i^{NN}\|_2$

end for

Find largest and smallest residual distances.

$d_{max}^{NN} = \max\{d_r(y, x_i^{NN})\}$

$d_{min}^{NN} = \min\{d_r(y, x_i^{NN})\}$

Step 4: Calculate the residual distance weights of k nearest neighbors.

for $i = 1$ to k **do**

if $d_{max}^{NN} \neq d_{min}^{NN}$ **then**

$wr_i = \frac{d_{max}^{NN} - d_r(y, x_i^{NN})}{d_{max}^{NN} - d_{min}^{NN}}$

else

$wr_i = 1$

end if

end for

Step 5: Classify y to the class c with the largest sum of weights among all the class.

$$l_y = \arg\max_{c} \sum_{(x_i^{NN}, l_i^{NN}) \in T_k} wr_i \times \tilde{\beta}_i \times \delta(c = l_i^{NN})$$

$y = \sum_{i=1}^{m} \bar{\beta}_i x_i$, $m \leq n$. Then, $\bar{\beta}_j$ of x_j can be derived as

$$\bar{\beta}_j = \frac{1}{2}\left(d^2(\sum_{i=1, i\neq j}^{m} \bar{\beta}_i x_i, x_j) - d^2(y, x_j)\right), \tag{21}$$

where $d(y, x_j)$ is Euclidean distance between y and x_j. Note that the derivation of Eq. (21) in details can be seen in [27]. We can see from Eq. (21) that the larger $d^2(\sum_{i=1, i\neq j}^{m} \bar{\beta}_i x_i, x_j)$ and/or the smaller $d^2(y, x_j)$ implies that x_j is much more likely to be a neighbor of y. Hence, x_j could be the true neighbor of y with a greater $\bar{\beta}_j$ with the integration of the larger $d^2(\sum_{i=1, i\neq j}^{m} \bar{\beta}_i x_i, x_j)$ and the smaller $d^2(y, x_j)$ simultaneously. In this case when $\bar{\beta}_i \neq 0$, x_i could be more likely to be a neighbor of y from the same class as y if $\bar{\beta}_i > 0$ and it could be from the different classes of y and not be a neighbor of y if $\bar{\beta}_i < 0$.

To visually demonstrate the fact above that sparse representation coefficients can be viewed as an effective similarity measure instead of the conventional Euclidean distance for seeking k nearest neighbors of the query samples, the examples of sparse coefficients of representing three query samples from three different classes on UCI Wine data set are given in Figs.1, 2 and 3. The information about UCI Wine data set is shown in Table 3, and it has 178 samples within three classes: the serial numbers of samples from 1 to 59 are within class 1, from 60 to 130 within class 2 and from 131 to 178 within class 3. These serial numbers corresponding to 178 samples are also clearly presented in Figs.1, 2 and 3. According to Eqs. (13) and (14), one query sample x_i from the samples X can be sparsely represented by the other remaining samples as

$$x_i = \alpha_1 x_1 + \alpha_2 x_2 +, \ldots, +\alpha_{i-1} x_{i-1} + \alpha_{i+1} x_{i+1} +, \ldots + \alpha_n x_n. \tag{22}$$

Accordingly, the reconstructive residual distance (residual for short) between x_i and x_j ($i \neq j$) can be calculated as

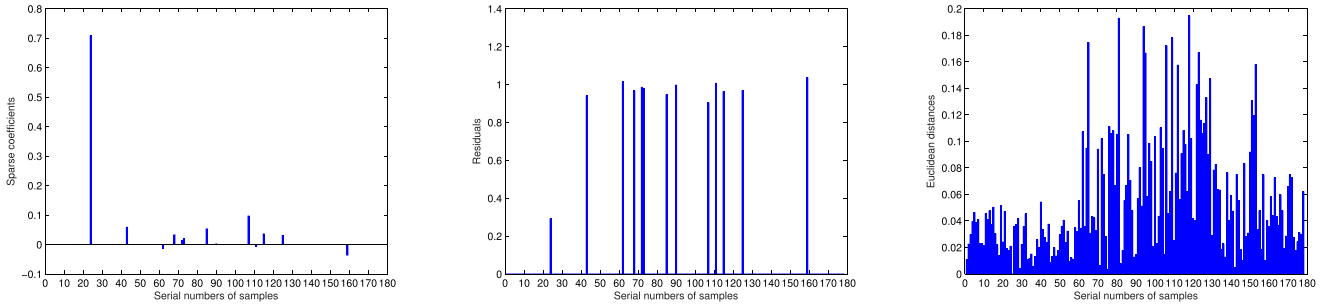
$$d_r(x_i, x_j) = \|x_i - \alpha_j x_j\|_2. \tag{23}$$

And the Euclidean distance between x_i and x_j is calculated as

$$d(x_i, x_j) = \|x_i - x_j\|_2. \tag{24}$$

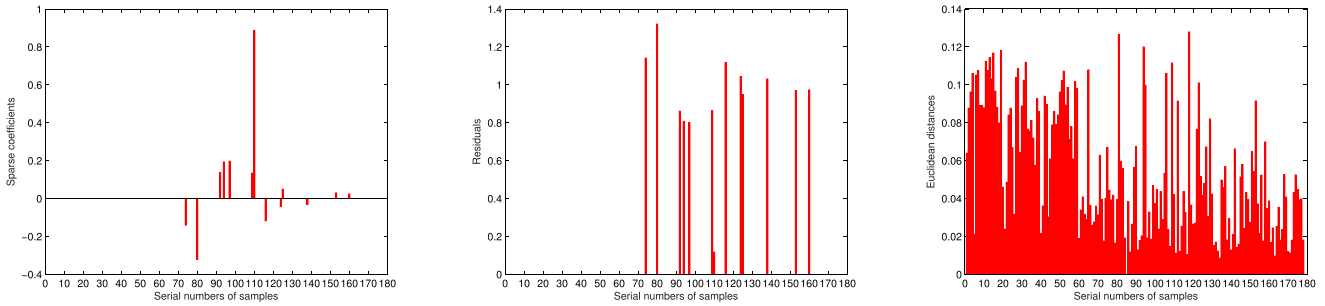
Then, in terms of Eqs. (22), (23) and (24), the sparse coefficients, residuals and Euclidean distances for the 25*th* sample in class 1, the 85*th* sample in class 2 and the 152*th* sample in class 3 on Wine are schematically depicted in Figs.1, 2 and 3, respectively. Note that when the serial numbers are 25 in Fig.1, 85 in Fig.2 and 152 in Fig.3, there are no corresponding coefficients, residuals and Euclidean distances. And also, since each sample in X has unit L_2-norm, when $\alpha_j = 0$ the residual between x_i and x_j is $d_r(x_i, x_j) = 1$, so the residuals with the values 1 are unnecessarily represented in these three figures.

As can be shown in Figs. 1(a), 2(a) and 3(a), the sparse coefficients of the samples from the same class as ith sample ($i = 25, 85, 152$) are always dominated, and the other ones of the samples from different classes are very small or tend to be zero in most cases. This experimental fact implies that sparse coefficients can be well adopted to choose the nearest neighbors of one sample from its same class. More importantly, from the residuals in Figs. 1(b), 2(b) and 3(b), the residuals of one sample from its same class are very smaller when the corresponding coefficients are larger. This phenomenon means that the larger coefficient of one sample to represent the other sample corresponds to the smaller residual. In other words, the larger coefficient and smaller residual of one sample indicate the more contribution to representing the other sample. It is also interesting to note that the residuals corresponding to the negative coefficients are very larger than or tend to be one, and the residuals corresponding



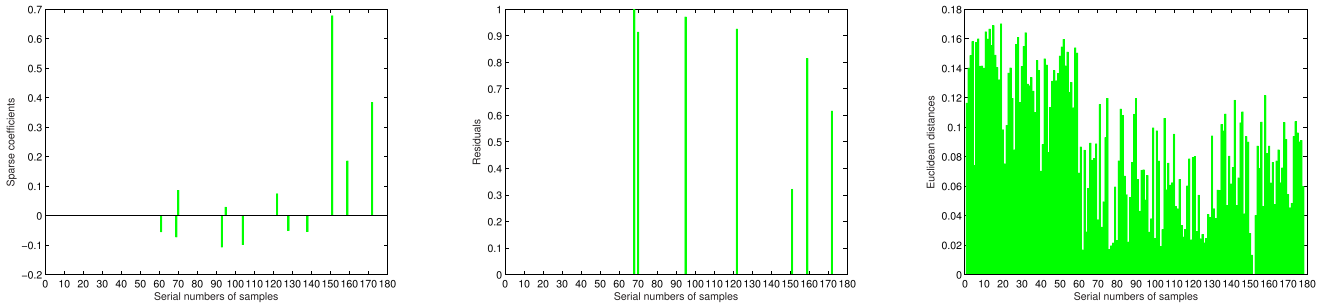
(a) (b) (c)

FIGURE 1. The sparse coefficients, residuals and Euclidean distances for the 25*th* sample in class 1 on UCI Wine data set. (a) Sparse coefficients. (b) Residuals. (C) Euclidean distances.



(a) (b) (c)

FIGURE 2. The sparse coefficients, residuals and Euclidean distances for the 85*th* sample in class 2 on UCI Wine data set. (a) Sparse coefficients. (b) Residuals. (C) Euclidean distances.



(a) (b) (c)

FIGURE 3. The sparse coefficients, residuals and Euclidean distances for the 152*th* sample in class 3 on UCI Wine data set. (a) Sparse coefficients. (b) Residuals. (C) Euclidean distances.

to the positive coefficients are smaller than one. This can confirm that the negative coefficients and the larger residuals means these corresponding samples are not representative in sparse representation. So the samples with the corresponding negative coefficients are not chosen as neighbors of each query sample. Compared to residuals, the Euclidean distances may not well represent the similarities among samples from the same and different classes, obviously shown in Figs. 1(c), 2(c) and 3(c). Therefore, the neighborhood selection of one sample using sparse coefficients can be more effective than the one using Euclidean distances. Furthermore, the experimental examples in Figs.1, 2 and 3 can also verify the fact that the sparse coefficients in sparse representation can well reflect the similarities of data [27], [29]–[31].

In terms of sparse coefficients, residuals in Figs.1, 2 and 3, the particular values of serial numbers, sparse coefficients and residuals of the first three nearest neighbors of *25*th, *85*th and *152*th samples with classification results by CWKNN and RWKNN are shown in Table 1. And according to the Euclidean distances in Figs. 1(c), 2(c) and 3(c), the particular values of serial numbers and Euclidean distances of the first three nearest neighbors of *25*th, *85*th and *152*th samples with classification results by KNN and WKNN are also shown in Table 2. As shown in Tables 1 and 2, we can observe that the first three nearest neighbors of one sample

TABLE 1. The examples of the classification results of CWKNN and RWKNN on UCI Wine data set. Wine has three classes that are denoted by the symbols "$\circ$", "+" and "$\diamond$" for classes 1, 2 and 3, respectively. The classes of samples are represented in parentheses. The symbols "$\times$" and "$\surd$" indicate the wrong and right classification, respectively.

Sample	Serial numbers, Sparse coefficients and residuals of 3 NNs				CWKNN	RWKNN
25*th* sample ($\circ$)	Serial numbers	24 ($\circ$)	107 (+)	43 ($\circ$)	$\circ$ $\surd$	$\circ$ $\surd$
	Coefficients	0.7091	0.0974	0.0589		
	Residuals	0.2914	0.9027	0.9412		
85*th* sample (+)	Serial numbers	110 (+)	97 (+)	94 (+)	+ $\surd$	+ $\surd$
	Coefficients	0.8883	0.1977	0.1937		
	Residuals	0.1185	0.8024	0.8080		
152*th* sample ($\diamond$)	Serial numbers	151 ($\diamond$)	172 ($\diamond$)	159 ($\diamond$)	$\diamond$ $\surd$	$\diamond$ $\surd$
	Coefficients	0.6783	0.3844	0.1857		
	Residuals	0.3219	0.6163	0.8151		

TABLE 2. The examples of the classification results of KNN and WKNN on UCI Wine data set. Wine has three classes that are denoted by the symbols "$\circ$", "+" and "$\diamond$" for classes 1, 2 and 3, respectively. The classes of samples are represented in parentheses. The symbols "$\times$" and "$\surd$" indicate the wrong and right classification, respectively.

Sample	Serial numbers and Euclidean distances of 3 NNs				KNN	WKNN
25*th* sample ($\circ$)	Serial numbers	75 (+)	29 ($\circ$)	142 ($\diamond$)	tie	+ $\times$
	Euclidean distances	0.0031	0.0043	0.0051		
85th sample (+)	Serial numbers	134 ($\diamond$)	163 ($\diamond$)	117 (+)	$\diamond$ $\times$	$\diamond$ $\times$
	Euclidean distances	0.0086	0.0098	0.0105		
152*th* sample ($\diamond$)	Serial numbers	151 ($\diamond$)	62 (+)	76 (+)	+ $\times$	$\diamond$ $\surd$
	Euclidean distances	0.0132	0.0168	0.0171		

chosen by sparse coefficients are nearly different from the ones by Euclidean distances. Moreover, the three nearest neighbors of one sample are nearly from the same class by sparse coefficients, but the three nearest neighbors of one sample by Euclidean distances are always from the different classes. We can also see from Table 1 that the first nearest neighbors of *25*th, *85*th and *152*th samples are from their same classes, respectively. Their first nearest neighbors have the largest coefficients and the smallest residuals among their their three nearest neighbors, as shown in Figs.1, 2 and 3. And the differences among the coefficients and the residuals of three nearest neighbors are very significant. Because of the superiority of the sparse coefficients for choosing the true nearest neighbors, the *25*th, *85*th and *152*th samples are easily and correctly classified into the right classes by CWKNN and RWKNN. However, most of the three nearest neighbors of the *25*th, *85*th and *152*th samples chosen by Euclidean distances are from their different classes, and the differences among Euclidean distances of their neighbors are not prominent. So that the *25*th, *85*th and *152*th samples are nearly mistakenly classified by KNN and WKNN. From the experimental examples above, it can be obviously observed that the sparse coefficients as a similarity measure have more benefits than the conventional Euclidean distance, as stated in [27] and [29]. Accordingly, the analysis above has fully proven that the proposed CWKNN and RWKNN can have more discrimination power than KNN and WKNN in many pattern classification problems.

In summary, the sparse representation coefficients can be well adopted to choose true nearest neighbors of one sample and applied to design more simple and effective classification decisions in pattern recognition. Thus, the proposed CWKNN and RWKNN have good classification performance that will be further verified in the next section V.

V. EXPERIMENTAL RESULTS

In this section, we conduct the extensive experiments on real UCI and KEEL data sets to demonstrate the effective classification performance of the proposed methods in terms of classification error or accuracy. We compare the proposed CWKNN and RWKNN with the state-of-the-art methods including SRC [23], SoC [31], LRMNN [27], LMPNN [5], WKNN, KNN, CRNN [41] and LPC [19]. For fair comparisons, the value of p in Eq. (9) is $p = 1$ for the LRMNN used in the experiments, because our proposed methods have the same regularization of the coefficients in Eq. (14) with the LRMNN. It should be noted that Eqs. (6), (9) and (14) are solved by using the CVX Matlab optimization toolbox (http://cvxr.com/cvx/download/) in the experiments.

A. DATA SETS

The real data sets used in the experiments are twenty UCI data sets from the UCI Repository [44], eight KEEL data sets without noises and four KEEL data sets with noises from the KEEL Repository [45]. Their main information about data sets is described in Table 3. These used data sets with different classes, attributes and samples only have numeric attributes in order that the classification of each competing method is easily done on them. Note that the data sets with long names are abbreviated. That is, Biodegradation, Cardiotocography, Image Segmentation, Ionosphere, Parkinsons, SteelPlates, Transfusion, Page-blocks, Wine quality red, Appendicitis, Wisconsin, Cleveland, Dermatology are abbreviated as Biodeg, Cardio, Image, Iono, Park, SteelP, Trans, Page, WineQR, Append, Wiscon, Cleve and Dermat, respectively. And the names of the Wine, Heart, Page-blocks and Thyroid data sets with attribute noises from the KEEL Repository are denoted as Wine_n, Heart_n, Page_n and Thyroid_n. To simplify the representation of all the data sets

TABLE 3. The real data sets used in the experiments.

No.	Datasets	Databases	Samples	Attributes	Classes	Testing samples
1	Biodeg	UCI	1055	41	2	365
2	Cardio	UCI	2126	21	10	710
3	Heart	UCI	267	44	2	92
4	ILPD	UCI	583	9	2	195
5	Image	UCI	2310	19	7	770
6	Iono	UCI	351	34	2	155
7	Park	UCI	195	22	2	65
8	Seed	UCI	210	7	3	60
9	Musk	UCI	6598	166	2	3098
10	SteelP	UCI	1941	27	7	646
11	Thyroid	UCI	7200	21	3	2485
12	Trans	UCI	748	4	2	130
13	Vehicle	UCI	846	18	4	285
14	Wdbc	UCI	569	30	2	192
15	Wine	UCI	178	13	3	59
16	Wpbc	UCI	198	32	2	58
17	Page	UCI	5473	10	5	1823
18	WineQR	UCI	1571	11	4	523
19	Plrx	UCI	182	12	2	40
20	Diabetic	UCI	1151	19	2	373
21	Tae	KEEL	151	5	3	60
22	Hayesroth	KEEL	160	4	3	60
23	Append	KEEL	106	7	2	34
24	Wiscon	KEEL	683	9	2	223
25	Cleve	KEEL	297	13	5	91
26	Band	KEEL	365	19	2	125
27	Saheart	KEEL	452	8	2	152
28	Dermat	KEEL	358	34	6	111
29	Wine_n	KEEL	178	13	3	59
30	Heart_n	KEEL	270	13	2	90
31	Page_n	KEEL	5472	10	5	1823
32	Thyroid_n	KEEL	7200	21	3	2485

in the experiments, each data set is indicated by the given corresponding serial number in Table 3.

In the experiments, each data set except the noise data sets is randomly divided into training and testing sets, shown in Table 3. We run each method on ten divisions of each data set for average classification evaluation with 95% confidence. In other words, we achieve the ten different training and testing sets on each data set and the average classification error or accuracy rates of each method are obtained in our experiments. It should be noted that all the experiments have been conducted by using Intel(R) Core(TM) i7-3770 CPU @3.40 GHz 3.40 GHz and 12 GB of memory in the 64-bit Windows 7 operating system.

B. EXPERIMENT 1

To verify the effectiveness of the proposed CWKNN and RWKNN methods, we carry out the extensive experiments on real UCI data sets and the KEEL ones without noises, in comparisons with SRC, SoC, LRMNN, LMPNN, KNN, WKNN, CRNN and LPC in this subsection.

The comparative classification performance of the proposed methods, CRNN, LPC, LMPNN, KNN and WKNN is first investigated by varying the numbers k of nearest neighbors in terms of the classification error. It should be noted that the neighborhood size k in CWKNN, RWKNN, CRNN, KNN and WKNN is for all the training samples of each query sample, while k is for the training samples from each class in the LMPNN and LPC. The range of the values of k is preseted from 1 to 11 with a step 1. The experimental classification results of these KNN-based methods via the neighborhood size k are displayed in Figs. 4 and 5. We can observe from the experimental results in Figs. 4 and 5 that the proposed CWKNN and RWKNN almost perform better than CRNN, LPC, LMPNN, KNN and WKNN with varying the numbers k of nearest neighbors, especially when the value of k is large. These experimental results indicate that the proposed methods are less sensitive to the neighborhood size k with promising classification performance. It is very interesting to note that the classification error rates of the proposed methods nearly decrease with increasing the values of k and then tend to be stable. This experimental fact implies that the proposed CWKNN and RWKNN can choose the suitable large numbers of the true neighbors of the query samples for the favorable classification. Thus, the proposed methods are robust to the neighborhood selection through sparse coefficients with the satisfactory performance.

To further evaluate the proposed methods, the maximal classification accuracy rates of each competing method with the standard deviations on all the data sets are shown in Table 4. Note that the best classification accuracy rates of CWKNN, RWKNN, CRNN, LMPNN, LPC, KNN and WKNN are obtained among the range of the neighborhood size k from 1 to 11 in terms of the classification results in Figs. 4 and 5. And the No. in Table 4 indicate the

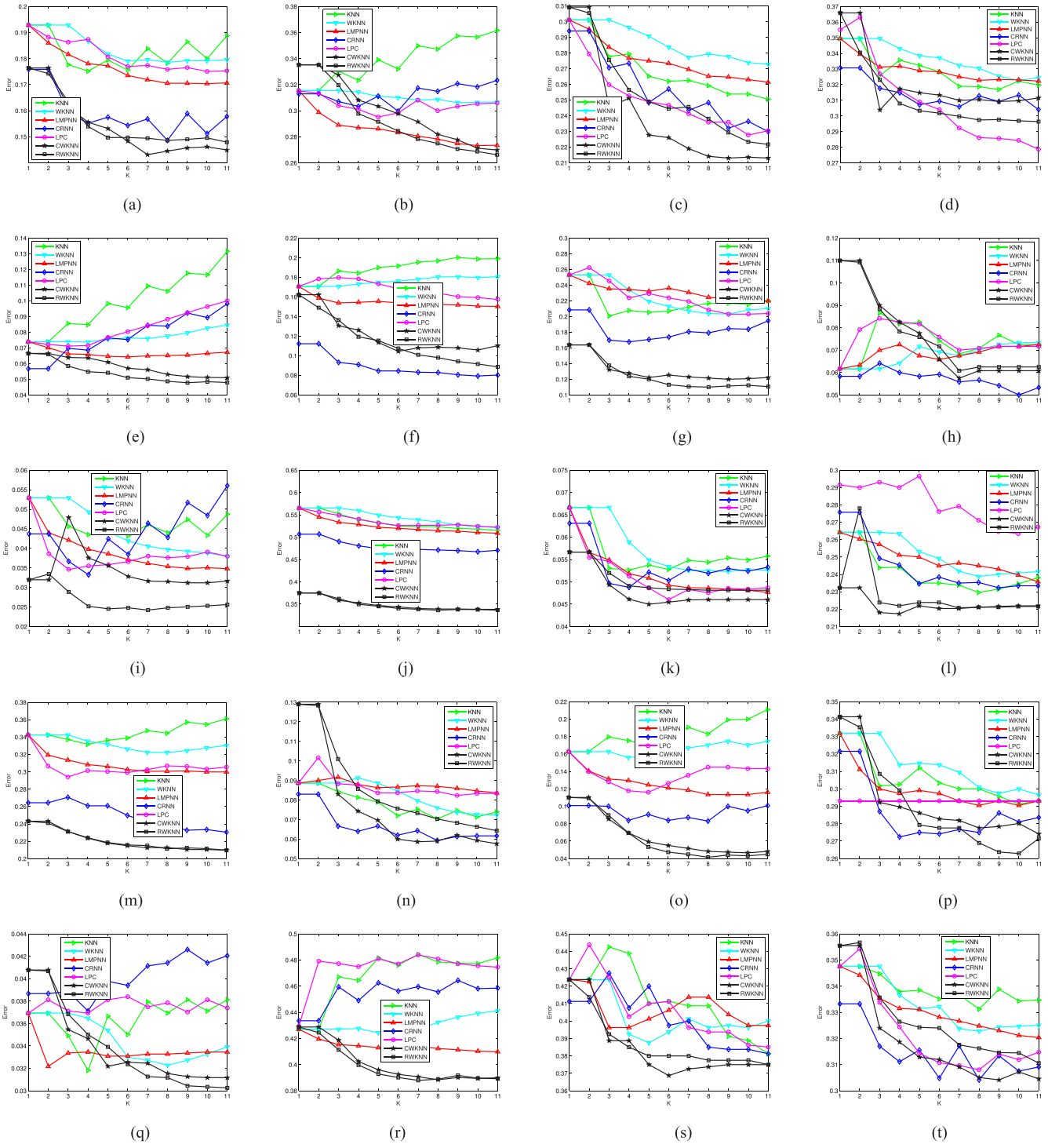


FIGURE 4. The classification error rates of the competing KNN-based methods via the neighborhood size k on the real UCI data sets. (a) Biodeg. (b) Cardio. (c) Heart. (d) ILPD. (e) Image. (f) Iono. (g) Park. (h) Seed. (i) Musk. (j) SteelP. (k) Thyroid. (l) Trans. (m) Vehicle. (n) Wdbc. (o) Wine. (p) Wpbc. (q) Page. (r) WineQR. (s) Plrx. (t) Diabetic.

corresponding data sets shown in Table 3. It is obvious that the proposed CWKNN and RWKNN outperform the other competing methods in most cases. CWKNN is better than SRC, SoC, LRMNN, LMPNN, CRNN, LPC, KNN and WKNN on 19 out of 28 data sets, and RWKNN is better than them on 20 out of 28 data sets. We can also see that the proposed methods perform better than LRMNN, LMPNN, CRNN, LPC, KNN and WKNN significantly on most of the data sets,

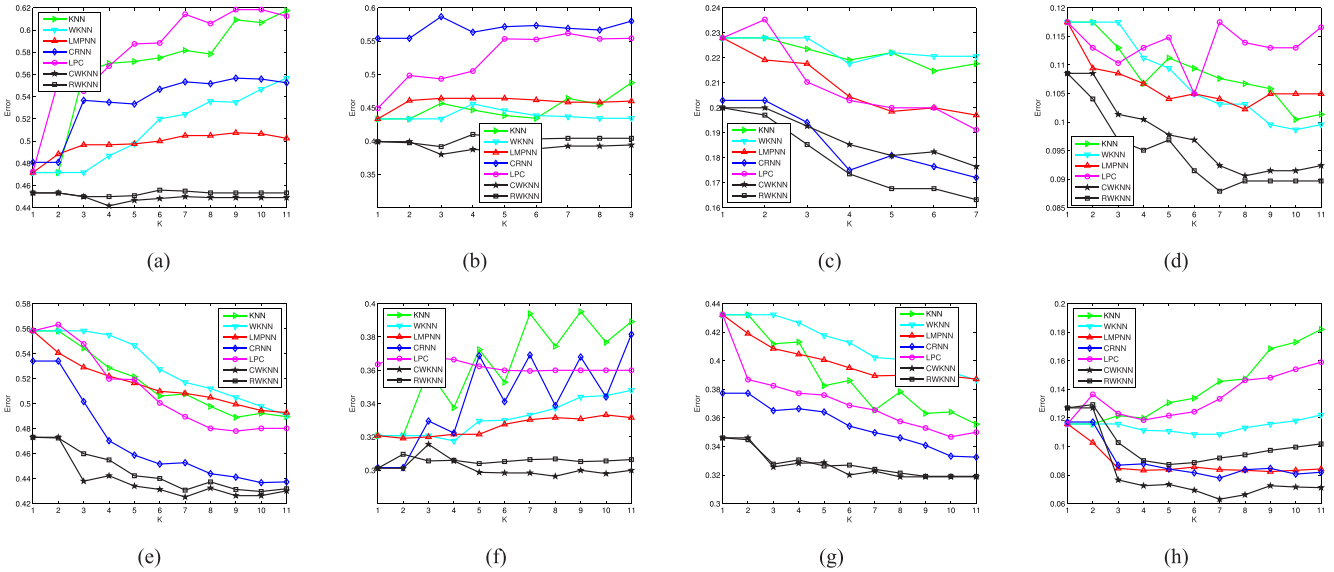


FIGURE 5. The classification error rates of the competing KNN-based methods via the neighborhood size k on the real KEEL data sets. (a) Tae. (b) Hayesroth. (c) Append. (d) Wiscon. (e) Cleve. (f) Band. (g) Saheart. (h) Dermat.

such as Park, StealP, Vehicle and Wine. Meanwhile, it has been found that the proposed CWKNN and RWKNN obtain the similar classification results as SRC and SoC in some degree. The reason can be that the CWKNN, RWKNN, SRC and SoC classifiers are based on the sparse representation of the query samples. Besides, the average classification accuracy rates of the proposed methods on all data sets are obviously larger than the ones of the other competing methods and their corresponding standard deviations are nearly smallest among them. This means the proposed methods are more stable in some degree. From the experimental results in Table 4, it can be concluded that the proposed CWKNN and RWKNN methods can choose the true nearest neighbors of the query samples for good classification.

Next, we employ one non-parametric statistical test, which is named as the Wilcoxon Signed-Ranks test, to make the comparisons of the competing methods in the experiments be statistically convincible and to further verify the good classification performance of the proposed CWKNN and RWKNN methods. According to the Wilcoxon Signed-Ranks test [46]–[48], the pairwise comparisons of the proposed methods with SRC, SoC, LRMNN, LMPNN, CRNN, LPC, KNN and WKNN should be first done using the classification results in Table 4. Let r_i denote the difference between the accuracy rates of any two methods on the ith out of N data sets. These pairwise difference r_i are sorted by their absolute values, and then the corresponding rank value of r_i is indicated as $rank(r_i)$. Meanwhile, let R^+ be the total sum of the ranks from the data sets on which one method performs better than the other one, and R^- the total sum of the ranks for the data sets on which one method performs worse than the other one. The average ranks are allocated in the case of ties, the ranks of $r_i = 0$ are divided evenly among the sums. Then, the sums of R^+ and R^- are computed as

$$R^+ = \sum_{r_i>0} rank(r_i) + \frac{1}{2}\sum_{r_i=0} rank(r_i),$$
$$R^- = \sum_{r_i<0} rank(r_i) + \frac{1}{2}\sum_{r_i=0} rank(r_i). \quad (25)$$

Let $R = min(R^+, R^-)$. When the number N of data sets is larger than 25, the statistics can be calculated

$$Z = \frac{R - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+1)}}, \quad (26)$$

and is distributed approximately normally. If $Z < -1.96$, the null-hypothesis should be rejected with $\alpha = 0.05$.

Using the Wilcoxon Signed-Ranks test and the classification results in Table 4, the pairwise comparisons of the proposed CWKNN and RWKNN with the other eight competing methods are achieved in Table 5. In Table 5, R^+ and R^- are the total sums of ranks where the proposed methods perform better and worse than the other competing method on each data set, respectively. It is obvious from Table 5 that R^+ is very larger than R^- and the statistics Z are too smaller than -1.96. This fact means that the differences between the proposed methods and the other competing methods are significant at $\alpha = 0.05$. As stated in [47], the significant information for a statistical hypothesis test is reflected by p-value and the smaller p-value gives the more evidence against the null hypothesis. According to the p-values in Table 5, we can clearly see that they are very smaller than $\alpha = 0.05$ and provide the significant differences between the proposed methods and the other competing methods. Therefore, the experimental results using the Wilcoxon

TABLE 4. The maximal classification accuracy rates (%) of each method with the corresponding standard deviations (stds) and values of k in the parentheses on all the data sets (the best recognition performance among competing methods on each data set is described in bold-face).

No.	SRC	SoC	LRMNN	KNN	WKNN	LMPNN	CRNN	LPC	CWKNN	RWKNN
1	85.04±1.70	84.84±1.52	82.01±1.90	82.48±1.43 (4)	82.14±1.65 (8)	82.96±2.05 (10)	85.14±1.98 (8)	82.49±1.86 (10)	**85.68±1.46 (7)**	85.21±1.67 (11)
2	72.93±0.43	72.96±0.53	60.45±1.42	68.45±2.24 (1)	69.38±1.58 (9)	72.68±1.23 (10)	70.03±2.29 (6)	70.48±1.82 (5)	73.01±0.75 (11)	**73.38±0.60 (11)**
3	71.09±3.79	70.76±3.73	67.50±3.61	74.95±3.35 (11)	72.72±2.61 (11)	73.86±3.62 (11)	77.01±1.66 (11)	77.23±2.85 (10)	**78.70±1.19 (9)**	77.83±1.91 (11)
4	69.74±2.43	66.82±3.33	61.18±3.78	68.31±2.35 (9)	67.72±2.60 (10)	67.77±3.07 (11)	69.59±2.48 (11)	**72.13±1.50 (11)**	69.62±3.34 (3)	70.36±3.39 (11)
5	95.19±0.58	95.14±0.59	94.95±0.59	92.61±0.90 (1)	92.62±0.76 (4)	93.57±0.61 (6)	94.32±0.58 (1)	92.88±0.86 (3)	94.90±0.88 (11)	**95.22±0.73 (9)**
6	90.74±1.86	80.94±3.09	83.73±2.94	82.94±1.97 (1)	82.94±1.97 (1)	84.97±1.85 (11)	**92.06±1.51 (10)**	84.26±2.78 (11)	89.55±1.91 (6)	91.13±2.00 (11)
7	86.31±4.49	86.46±4.25	77.31±5.23	79.92±3.78 (3)	79.69±3.76 (9)	78.00±3.77 (11)	83.23±3.53 (4)	79.69±3.33 (9)	88.00±2.85 (9)	**89.00±2.65 (8)**
8	92.83±3.06	92.92±3.00	89.58±3.54	93.83±3.02 (1)	93.83±3.02 (1)	93.83±3.02 (1)	**95.00±1.95 (10)**	93.83±3.02 (1)	94.25±2.13 (7)	93.92±2.04 (7)
9	**97.85±0.29**	97.77±0.22	96.77±0.35	95.68±0.19 (11)	96.20±1.57 (11)	96.52±1.43 (4)	96.68±0.18 (3)	96.54±0.12 (9)	96.88±1.28 (7)	97.58±1.02 (6)
10	65.82±1.78	66.03±1.72	61.82±1.69	48.92±1.37 (11)	47.82±1.16 (11)	49.11±1.27 (11)	53.26±1.35 (10)	47.79±0.99 (11)	66.16±1.38 (11)	**66.35±1.49 (8)**
11	95.03±0.30	95.06±0.26	94.33±0.52	94.74±0.16 (4)	94.77±0.14 (9)	95.24±0.13 (11)	95.12±0.23 (4)	95.40±0.22 (6)	**95.50±0.12 (5)**	95.19±0.35 (8)
12	72.85±10.98	76.96±3.23	76.73±2.97	77.04±2.52 (8)	76.12±3.31 (8)	76.42±2.61 (11)	76.77±2.11 (9)	73.65±2.90 (10)	**78.27±2.49 (4)**	77.92±2.69 (7)
13	78.42±2.06	78.37±2.11	76.26±1.66	66.84±2.05 (4)	67.77±2.49 (8)	70.02±2.10 (11)	76.95±2.14 (11)	70.60±2.58 (3)	**79.02±2.01 (11)**	**79.02±1.83 (11)**
14	**95.50±1.66**	**95.50±1.66**	89.63±1.49	92.97±1.82 (8)	92.77±1.64 (11)	91.63±1.47 (11)	94.07±1.63 (8)	91.77±1.47 (9)	94.23±1.84 (11)	93.57±1.81 (11)
15	94.24±2.55	94.15±3.13	90.00±2.90	83.73±6.45 (1)	84.41±5.65 (4)	88.64±4.26 (8)	91.69±4.29 (8)	88.39±3.86 (5)	95.34±2.32 (10)	**95.85±2.55 (8)**
16	70.34±4.83	71.98±4.18	65.69±4.44	70.86±1.24 (10)	70.34±2.77 (11)	70.95±2.76 (8)	72.76±3.19 (4)	70.69±0.00 (1)	72.59±2.23 (11)	**73.71±3.58 (10)**
17	96.75±0.18	96.75±0.18	95.92±0.38	96.82±0.15 (4)	96.77±0.21 (8)	96.78±0.35 (2)	96.29±0.39 (4)	96.31±0.41 (4)	96.88±0.24 (10)	**96.97±0.21 (11)**
18	60.76±1.66	61.04±1.54	57.31±1.70	57.29±1.11 (1)	57.75±1.51 (6)	59.02±1.33 (11)	56.64±1.67 (1)	57.06±1.08 (1)	61.16±1.63 (8)	**61.20±1.84 (7)**
19	60.50±6.16	50.38±7.79	56.75±6.18	61.88±1.79 (11)	61.25±5.53 (5)	60.38±4.39 (3)	61.88±1.97 (11)	61.50±1.70 (11)	**63.13±4.58 (6)**	62.50±4.21 (11)
20	69.14±2.16	69.30±2.12	64.01±2.08	66.88±1.87 (8)	67.72±2.00 (8)	67.96±1.62 (11)	**69.58±1.78 (8)**	69.20±1.87 (8)	**69.58±2.23 (9)**	68.94±2.24 (11)
21	53.67±5.09	53.50±4.92	54.67±5.66	52.83±5.85 (1)	52.83±5.85 (1)	52.83±5.85 (1)	51.92±5.55 (1)	52.75±6.05 (1)	**55.83±5.86 (4)**	55.00±5.49 (4)
22	58.83±6.58	59.08±6.52	60.08±6.45	56.67±6.60 (1)	56.67±6.60 (1)	56.67±6.60 (1)	44.58±7.53 (1)	55.08±6.03 (1)	**62.00±6.43 (3)**	60.83±6.57 (3)
23	82.35±5.73	82.65±5.22	80.29±5.73	78.53±3.83 (6)	78.24±5.26 (4)	80.29±3.71 (7)	82.79±3.20 (7)	80.88±3.51 (7)	82.35±4.27 (7)	**83.68±4.72 (7)**
24	90.49±1.89	90.40±1.57	89.15±1.07	89.96±1.40 (10)	90.13±1.74 (10)	89.78±0.86 (8)	61.17±1.97 (6)	89.51±1.44 (6)	90.94±2.14 (8)	**91.21±2.05 (7)**
25	56.04±2.85	56.10±2.84	49.95±4.03	51.10±3.06 (9)	50.93±3.50 (11)	50.71±3.29 (11)	56.32±2.76 (10)	52.20±2.91 (9)	**57.47±3.86 (7)**	57.03±3.29 (10)
26	69.40±2.03	69.72±1.80	65.40±4.41	67.92±3.99 (1)	68.24±3.41 (4)	68.08±3.89 (2)	69.84±3.52 (1)	64.04±0.18 (7)	**70.36±1.93 (8)**	69.88±1.91 (1)
27	67.57±3.15	67.53±2.98	65.20±3.49	64.44±2.95 (11)	61.35±2.66 (11)	61.28±2.51 (11)	66.74±3.15 (11)	65.33±3.98 (10)	**68.13±3.23 (8)**	68.09±2.97 (9)
28	89.23±2.52	**95.41±2.04**	85.59±3.46	88.42±2.65 (1)	89.14±3.13 (6)	91.76±2.79 (9)	92.21±2.97 (7)	88.42±2.70 (1)	93.69±2.15 (7)	91.26±2.32 (5)
Mean	78.17±2.96	77.80±2.72	74.72±2.99	75.25±2.50	75.08±2.79	75.78±2.59	76.20±2.41	75.72±2.22	**79.41±2.38**	79.35±2.43

Signed-Ranks test consistently demonstrate the better classification performance of the proposed methods, compared to SRC, SoC, LRMNN, LMPNN, CRNN, LPC, KNN and WKNN.

C. EXPERIMENTS 2

In this subsection, we further verify the effectiveness of the proposed methods by conducting the comparative experiments on four KEEL attribute noise data sets shown in Table 3. Note that the four noise data sets have been downloaded in the KEEL Repository. These noise data sets are three types of noise data sets: 5% Noisy Train-Noisy Test, 5% Noisy Train-Clean Test and 5% Clean Train- Noisy Test. That is to say, '5% Noisy Train-Noisy Test' is with 5% of noise in training and testing sets, '5% Noisy Train-Clean Test' is with 5% of noise in training set while testing one remains unchanged, and ' 5% Clean Train-Noisy Test' is with 5% of noise in testing set while training one remains unchanged. That is, the 5% of the samples in each training or testing set are randomly selected and artificially corrupted by the noises [45]. Thus, these four noise data sets can be regarded as the simulated data. These three types of noise data sets are denoted by n-n, n-c and c-n, respectively. For the n-n data sets, the training and testing samples are

TABLE 5. The pairwise comparisons of the proposed methods with SRC, SoC, LRMNN, KNN, WNN, LMPNN, CRNN and LPC on all the numerical data sets using the Wilcoxon Signed-Ranks test ('True' denotes the significant difference between the classification performance of any two comparative methods).

Pairwise Comparsion	R^+	R^-	Statistics(Z)	*p-value*	Significant Difference
CWKNN vs. SRC	347.5	58.5	-3.2905	5.000×10^{-4}	True
CWKNN vs. SoC	340	66	-3.1197	9.052×10^{-4}	True
CWKNN vs. LRMNN	405	1	-4.5998	2.114×10^{-6}	True
CWKNN vs. KNN	406	0	-4.6226	1.895×10^{-6}	True
CWKNN vs. WKNN	406	0	-4.6226	1.895×10^{-6}	True
CWKNN vs. LMPNN	406	0	-4.6226	1.895×10^{-6}	True
CWKNN vs. CRNN	362.5	43.5	-3.6320	1.406×10^{-4}	True
CWKNN vs. LPC	390	13	-4.3266	7.571×10^{-6}	True
RWKNN vs. SRC	375	31	-3.9167	4.488×10^{-5}	True
RWKNN vs. SoC	346	60	-3.2563	5.644×10^{-4}	True
RWKNN vs. LRMNN	406	0	-4.6226	1.895×10^{-6}	True
RWKNN vs. KNN	406	0	-4.6226	1.895×10^{-6}	True
RWKNN vs. WKNN	406	0	-4.6226	1.895×10^{-6}	True
RWKNN vs. LMPNN	401	5	-4.5087	3.261×10^{-6}	True
RWKNN vs. CRNN	349	57	-3.3246	4.427×10^{-4}	True
RWKNN vs. LPC	392	14	-4.3038	8.395×10^{-6}	True

TABLE 6. The maximal classification accuracy rates (%) of each method with the corresponding standard deviations (stds) and values of k in the parentheses on four KEEL attribute noise data sets (the best recognition performance is described in bold-face on each data set).

No	SRC	SoC	LRMNN	KNN	WKNN	LMPNN	CRNN	LPC	CWKNN	RWKNN
29	87.29±3.32	87.46±3.41	80.00±6.13	77.80±7.43	78.64±5.37	80.85±4.31	87.46±3.50	81.53±4.11	**89.32±4.59**	88.81±4.24
(n-n)				(1)	(5)	(9)	(8)	(8)	**(9)**	(9)
30	74.56±2.89	74.44±2.87	69.44±4.33	68.33±4.81	66.33±4.32	65.33±4.82	74.11±3.36	68.00±3.58	**75.33±3.30**	74.33±2.37
(n-n)				(11)	(10)	(11)	(10)	(11)	**(3)**	(9)
31	93.92±0.80	93.92±0.82	92.67±0.81	94.02±0.27	94.11±0.25	94.22±0.34	93.93±0.22	93.80±0.43	**94.56±0.51**	94.20±0.67
(n-n)				(6)	(11)	(9)	(6)	(8)	**(5)**	(7)
32	92.38±0.18	92.52±0.17	91.35±0.26	93.29±0.03	93.04±0.04	93.20±0.17	93.31±0.16	93.09±0.18	**93.55±0.23**	92.78±0.20
(n-n)				(6)	(10)	(10)	(6)	(11)	**(4)**	(6)
29	91.52±5.39	91.52±5.39	88.71±7.31	81.43±4.45	81.40±6.42	84.21±6.39	93.22±3.31	87.05±3.35	93.78±4.25	**93.79±3.17**
(n-c)				(5)	(5)	(11)	(11)	(3)	(7)	**(9)**
30	76.30±4.79	76.67±4.46	70.00±4.61	70.74±5.77	70.00±4.22	67.04±6.60	**78.15±7.68**	70.00±5.62	76.67±2.48	77.04±2.81
(n-c)				(9)	(11)	(11)	**(10)**	(3)	(7)	(8)
31	96.45±0.47	96.47±0.46	95.78±0.69	96.49±0.58	96.67±0.49	**96.84±0.37**	96.31±0.42	96.38±0.35	96.47±0.25	96.53±0.36
(n-c)				(4)	(7)	**(10)**	(4)	(9)	(6)	(6)
32	94.24±0.27	94.22±0.30	93.76±0.54	94.01±0.43	94.33±0.41	**94.71±0.50**	94.28±0.22	94.46±0.35	94.64±0.28	94.42±0.31
(n-c)				(4)	(8)	**(8)**	(6)	(5)	(4)	(4)
29	90.40±6.60	90.40±6.60	82.54±4.81	81.95±9.22	81.43±6.91	84.22±6.59	89.30±4.70	84.79±5.94	91.54±3.57	**91.56±2.08**
(c-n)				(4)	(11)	(10)	(11)	(3)	(7)	**(6)**
30	**78.52±4.26**	78.15±4.79	69.26±5.33	70.37±4.14	69.63±8.03	67.04±7.57	75.93±5.86	67.41±5.00	77.04±4.46	76.67±4.65
(c-n)				(11)	(11)	(10)	(11)	(8)	(10)	(10)
31	79.97±2.14	80.79±1.47	81.18±5.24	87.04±0.51	87.30±0.33	**87.17±1.00**	83.42±0.99	86.93±1.03	86.37±2.08	86.75±0.42
(c-n)				(8)	(10)	**(11)**	(9)	(11)	(11)	(11)
32	91.49±0.84	91.43±0.82	90.64±0.52	92.00±0.20	91.79±0.27	91.22±0.37	91.15±0.37	91.49±0.36	**92.97±0.19**	92.38±0.39
(c-n)				(10)	(10)	(11)	(6)	(10)	**(7)**	(7)
Mean	87.25±2.66	87.33±2.63	83.78±3.38	83.96±3.15	83.72±3.09	83.84±3.25	87.55±2.57	84.58±2.53	**88.52±2.18**	88.27±1.81

randomly divided by 10 times and the size of testing samples is 59 on Wine_n, 90 on Heart_n, 1823 on Page_n and 2485 on Thyroid_n, shown in Table 3. The classification results of each method are the averages of ten runs on each noise data set with 95% confidence. For the n-c and c-n data sets, they have been already partitioned by means of a 5-folds cross validation procedure in the KEEL Repository [45]. The classification classification results of each method are the averages of five runs on each noise data set with 95% confidence.

On the four noise data sets, we first conduct the comparative experiments of the competing KNN-based methods with varying the numbers k of nearest neighbors of the query samples in terms of the classification error rate. The range of the neighborhood size k is also from 1 to 11 with a step 1. The classification results of the methods via the neighborhood size k are illustrated in Fig. 6. We can observe that the proposed CWKNN and RWKNN methods obtain the better classification performance than KNN, WKNN, CRNN, LPC and LMPNN in most cases with different values of k. Furthermore, the classification error rates of the proposed methods on these four noise data sets nearly decrease with the increase of k and then tend to be stable. Hence, we can conclude that the proposed methods have less sensitiveness and more robustness to k than KNN, WKNN, CRNN, LPC and LMPNN with satisfactory classification performance in the cases of noise.

The maximal average classification accuracy rates of each method on the four noise data sets are shown in Table 6. Note that the best classification performance of KNN, WKNN, CRNN, LPC, LMPNN, CWKNN and RWKNN is achieved at the interval of neighborhood size k ranged from 1 to 11.

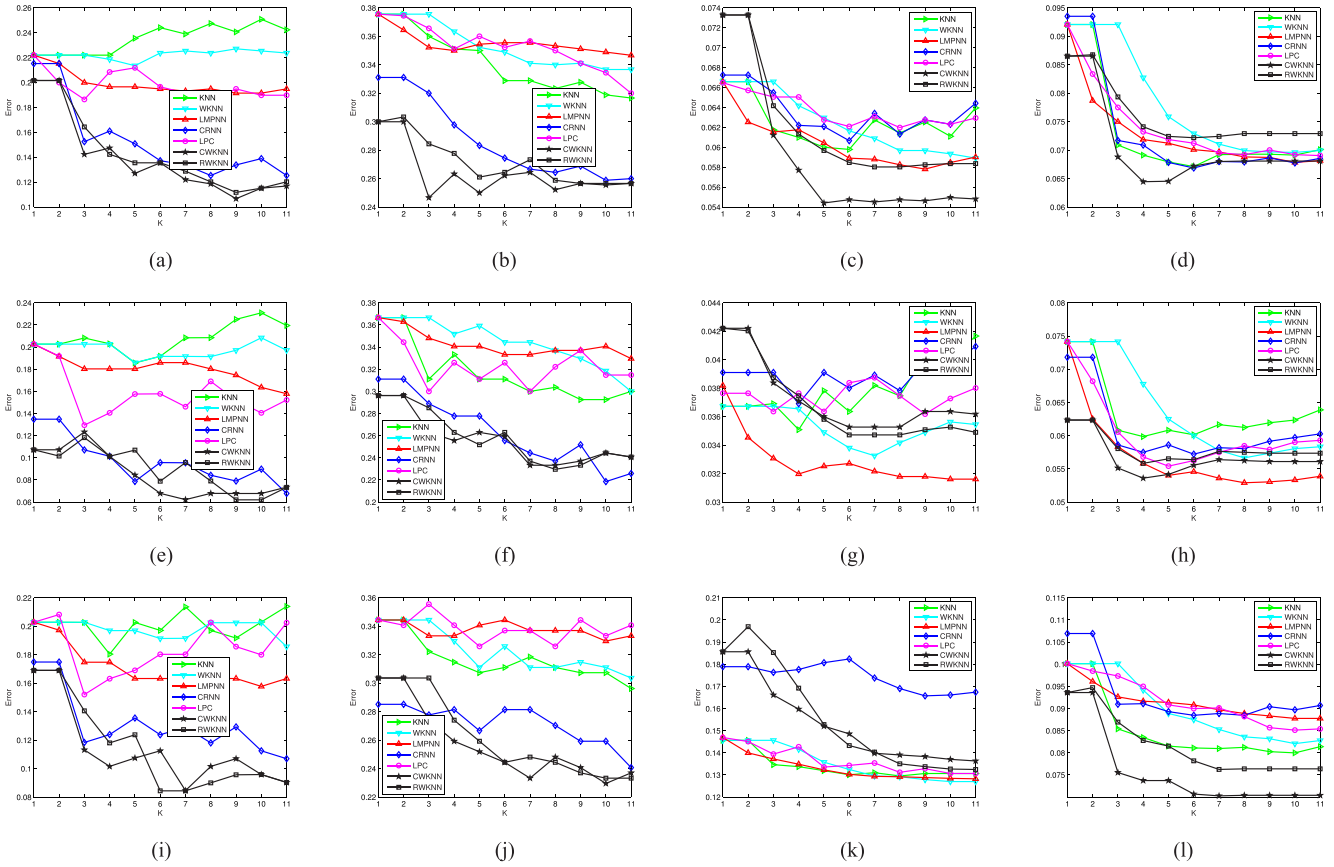


FIGURE 6. The classification error rates of the competing KNN-based methods via the neighborhood size k on four real KEEL data sets with attribute noises. (a) Wine_n (n-n). (b) Heart_n (n-n). (c) Page_n (n-n). (d) Thyroid_n (n-n). (e) Wine_n (n-c). (f) Heart_n (n-c). (g) Page_n (n-c). (h) Thyroid_n (n-c). (i) Wine_n (c-n). (j) Heart_n (c-n). (k) Page_n (c-n). (l) Thyroid_n (c-n).

We can see from the experimental results in Table 6 that the proposed methods obtain the better or comparative classification performance, compared to SRC, SoC, KNN, WKNN, LRMNN, CRNN, LPC and LMPNN. Furthermore, the average classification accuracy rates of the proposed methods are obviously larger than the other competing methods on these noise data sets and their standard deviations are smaller than the other methods. Therefore, the effectiveness of the proposed methods are also demonstrated on the noise data sets.

In summary, through the extensive experiments above, several observations can be concluded as follows:

(a) The proposed methods have less sensitiveness than the competing KNN-based classifiers with the different values of the neighborhood size k. And their robustness with good classification is evident when the values of k are large.

(b) The proposed methods always achieve the best classification performance among the competing methods, so that they are effective algorithms in many pattern classification problems.

(c) The effectiveness and robustness of the proposed methods further implies the sparse coefficients in sparse representation can be well employed to choose the representative nearest neighbors of one sample in pattern recognition, such as in the KNN-based classification.

VI. DISCUSSIONS

In this section, we first clearly emphasize the differences between the proposed CWKNN method and the closely related LRMNN and CRNN methods, and then provide the computational complexities of all the competing classification methods.

A. DIFFERENCES BETWEEN CWKNN AND LRMNN, CRNN

From the point view of L_1-norm constraint of the representation coefficients, we compare the proposed CWKNN with LRMNN (*i.e.* $p = 1$ in Eq. (9)). Meanwhile, CWKNN is further compared to CRNN in which representation coefficients are constrained with L_2-norm regularization.

a) CWKNN v.s. LRMNN: In LRMNN with L_1-norm regularization, the only one representative nearest neighbor of the query sample y is determined by the largest optimal value $\|\bar{a}_i\|$ of the representation coefficient a_i associated with the training sample x_i in Eq. (9). That is to say, the representative nearest neighbor of y is x_i, and then the query sample of y is classified into the class label

TABLE 7. The run time (s) of each competing method on Wine, Heart, Dermat and Vehicle (the values in the parentheses indicate the numbers of training samples, testing samples, attributes and classes, respectively).

Data	SRC	SoC	LRMNN	KNN	WKNN	LMPNN	CRNN	LPC	CWKNN	RWKNN
Wine (178,59,13,3)	4.4024	3.6993	3.6723	0.0054	0.0075	0.0131	0.0742	0.0253	3.7141	3.7175
Heart (267,92,44,2)	8.2152	7.5363	7.4983	0.0101	0.0122	0.0193	0.2949	0.0402	7.5714	7.6136
Dermat (358,111,34,6)	8.7873	8.0574	7.8710	0.0113	0.0145	0.0369	0.8261	0.0828	7.8438	7.8465
Vehicle (846,285,18,4)	25.7054	24.7116	24.6784	0.0290	0.0331	0.0732	19.6762	0.1601	24.6250	24.7905

of x_i by using Eq. (10). Just like LRMNN with L_1-norm constraint of representation coefficients, CWKNN utilizes the same objective function to optimize the representation coefficients (*i.e.* Eq. (9) when $p = 1$ is the same as Eq. (14)). However, unlike LRMNN, the proposed CWKNN employs the top k largest sparse representation coefficients to find k nearest neighbors of the query sample y, the training samples corresponding to the top k largest coefficients are k nearest neighbors of y. And then, the coefficients of k nearest neighbors in CWKNN are viewed as their corresponding weights to represent and classify y, finally the class of y is determined by the weighted majority voting in Eq. (15). Thus, we can easily see that the ways of choosing nearest neighbors and the classification decisions used in LRMNN and CWKNN are different.

b) CWKNN v.s. CRNN: CRNN and CWKNN use different constraints of the representation coefficients. In CRNN, the representation coefficients with L_2-norm regularization are weighted by the biasing Tikhonov matrix in Eq. (12), and the class label of the query sample y is determined by the majority voting in Eq. (2) among the k nearest neighbors corresponding to k largest coefficients. However, in CWKNN the representation coefficients are obtained by L_1-norm regularization in Eq. (14), and the class label of the query sample y is determined by the weighted majority voting among k nearest neighbors corresponding to k largest coefficients in Eq. (15) and each nearest neighbor is weighted by the value of the corresponding representation coefficient. Thus, we can also clearly see that the ways of obtaining the representation coefficients and the classification decisions used in CRNN and CWKNN are different.

B. ANALYSIS OF COMPUTATIONAL COMPLEXITY

To embody the efficiency of the proposed methods in comparisons with the related competing methods, we further analyze the computational complexities of SRC, SoC, LRMNN, KNN, WKNN, LMPNN, CRNN, LPC, LRMNN and the proposed CWKNN and RWKNN. Note that we first specify the notations in the computational complexities as follows: n, d and k denote the numbers of all training samples, attributes of each sample and the nearest neighbors, respectively.

For a query sample in KNN, WKNN, LMPNN, the major computational time is to compute the Euclidean distances between the query sample and all training samples and to find k nearest neighbors, so the main computational complexities of KNN, WKNN and LMPNN are $O(nd + nk + k)$, $O(nd + nk + 2k)$ and $O(nd + nk + 2ckd + 3ck + c)$, respectively [5]. In LPC [19], the major computational time of classifying a query sample is to find k_0 nearest neighbors of each training sample and k nearest neighbors of the query sample through the Euclidean distance measure, so the main computational complexity is $O(n^2d + n^2k_0 + nd + nk + ckd + cd + c)$. In CRNN, the computational time of classifying a query sample is to find k nearest neighbors of the query sample by first solving the Eq. (11), so the computational complexity is about $O(n^3 + nd + k)$. For SRC, SoC, LRMNN and the proposed CWKNN and RWKNN, the major computational time of classifying a query sample is to find k nearest neighbors of the query sample by solving the L_1 minimization problems, so the computation complexities of SRC, SoC, LRMNN, CWKNN and RWKNN are about $O(\varsigma n^3 + nc + c)$, $O(\varsigma n^3 + nc + 2c)$, $O(\varsigma n^3 + n)$, $O(\varsigma n^3 + nk + k)$, $O(\varsigma n^3 + nk + kd + 2k)$, respectively. And ς is the number of iterations to achieve the optimal representation coefficients in the L_1 minimization problems.

To further intuitively show the computational complexities of the competing classification methods, The run time of each method, for example, on the Wine, Heart, Dermat and Vehicle data sets with different numbers of training samples, testing samples, attributes and classes is displayed in Table 7. We can observe that the experimental time of all the methods fits their theoretical computational complexities.

VII. CONCLUSIONS

In this article, we present two novel weighted KNN classifiers, called the coefficient-weighted k-nearest neighbor method (CWKNN) and the residual-weighted k-nearest neighbor method (RWKNN). In the proposed methods, the sparse representation coefficients of all the training samples to represent each query sample using sparse representation are adopted to choose k representative nearest neighbors. In the CWKNN, k nearest neighbors of each query sample are selected by sparse coefficients, and the representation coefficient of each neighbor as its weight is utilized for the weighted majority voting in the KNN classification decision.

In the RWKNN, k nearest neighbors of each query sample are selected by sparse coefficients, and the reconstructive residuals between neighbors and the query sample are then calculated, finally the residual-weighted voting method is designed with integrating the sparse coefficients for the KNN classification decision. The classification performance of the proposed methods is studied by conducting the extensive experiments on UCI and KEEL data sets, compared to the competing methods including SRC, SoC, KNN, WKNN, LRMNN, CRNN, LPC and LMPNN. The experimental classification results have demonstrated the effectiveness and robustness of proposed CWKNN and RWKNN methods. Thus, the proposed methods are the promising algorithms for classification. In the future work, we plan to use the proposed methods for some practical applications in pattern recognition.

REFERENCES

[1] X. Wu *et al.*, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2008.

[2] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. IT-13, no. 1, pp. 21–27, Jan. 1967.

[3] T. Wagner, "Convergence of the nearest neighbor rule," *IEEE Trans. Inf. Theory*, vol. IT-17, no. 5, pp. 566–571, Sep. 1971.

[4] J. Gou, L. Du, Y. Zhang, and T. Xiong, "A new distance-weighted k-nearest neighbor classifier," *J. Inf. Comput. Sci.*, vol. 9, no. 6, pp. 1429–1436, 2012.

[5] J. Gou, Y. Zhan, Y. Rao, X. Shen, X. Wang, and W. He, "Improved pseudo nearest neighbor classification," *Knowl-Based Syst.*, vol. 70, pp. 361–375, Nov. 2014.

[6] G.-P. Nicolás, A. R. del C. Juan, and C.-G. Gonzalo, "A proposal for local k values for k-nearest neighbor rule," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 2, pp. 470–475, Feb. 2017.

[7] B. B. Chaudhuri, "A new definition of neighborhood of a point in multi-dimensional space," *Pattern Recognit. Lett.*, vol. 17, no. 1, pp. 11–17, 1996.

[8] J. S. Sáncheza, F. Plaa, and F. J. Ferri, "On the use of neighbourhood-based non-parametric classifiers," *Pattern Recognit. Lett.*, vol. 18, pp. 1179–1186, Nov. 1997.

[9] H. Altinçay, "Improving the k-nearest neighbour rule: Using geometrical neighbourhoods and manifold-based metrics," *Expert Syst.*, vol. 28, no. 4, pp. 391–406, 2011.

[10] J. Gou, Y. Zhang, L. Du, and T. Xiong, "A local mean-based k-nearest centroid neighbor classifier," *Comput. J.*, vol. 55, no. 9, pp. 1058–1071, 2012.

[11] R. Paredes and E. Vidal, "Learning weighted metrics to minimize nearest-neighbor classification error," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 7, pp. 1100–1110, Jul. 2006.

[12] K. Q. Weinberger and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," *J. Mach. Learn. Res.*, vol. 10, pp. 207–244, Feb. 2009.

[13] S. A. Dudani, "The distance-weighted k-nearest neighbor rule," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-6, no. 4, pp. 325–327, Apr. 1976.

[14] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. San Diego, CA, USA: Academic, 1990.

[15] Y. Mitani and Y. Hamamoto, "A local mean-based nonparametric classifier," *Pattern Recognit. Lett.*, vol. 27, no. 10, pp. 1151–1159, 2006.

[16] Y. Zeng, Y. Yang, and L. Zhao "Pseudo nearest neighbor rule for pattern classification," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3587–3595, 2009.

[17] S. Liang *et al.*, "Feature selection and predictors of falls with foot force sensors using KNN-based algorithms," *Sensors*, vol. 15, no. 11, pp. 29393–29407, 2015.

[18] Z. Pan, Y. Wang, and W. Ku, "A new k-harmonic nearest neighbor classifier based on the multi-local means," *Expert Syst. Appl.*, vol. 67, pp. 115–125, Jan. 2017.

[19] B. Li, Y. W. Chen, and Y. Q. Chen, "The nearest neighbor algorithm of local probability centers," *IEEE Trans. Syst. Man, Cybern. B, Cybern*, vol. 38, no. 1, pp. 141–154, Feb. 2008.

[20] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[21] D. L. Donoho, "For most large underdetermined systems of linear equations the minimal ℓ^1-norm solution is also the sparsest solution," *Commun. Pure Appl. Math.*, vol. 59, no. 6, pp. 797–829, 2006.

[22] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, "A survey of sparse representation: Algorithms and applications," *IEEE Access*, vol. 3, pp. 490–530, 2015.

[23] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[24] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," in *Proc. CVPR*, Colorado Springs, CO, USA, Jun. 2011, pp. 625–632.

[25] J. Wang, C. Lu, M. Wang, P. Li, S. Yan, and X. Hu, "Robust face recognition via adaptive sparse representation," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2368–2378, Dec. 2014.

[26] M. Yang, L. Zhang, X. Feng, and D. Zhang, "Sparse representation based Fisher discrimination dictionary learning for image classification," *Int. J. Comput. Vis.*, vol. 109, no. 3, pp. 209–232, 2014.

[27] J. Zhang and J. Yang, "Linear reconstruction measure steered nearest neighbor classification framework," *Pattern Recognit.*, vol. 47, no. 4, pp. 1709–1720, 2014.

[28] J. Zou, W. Li, and Q. Du, "Sparse representation-based nearest neighbor classifiers for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 12, pp. 2418–2422, Dec. 2015.

[29] H. Cheng, Z. Liu, L. Hou, and J. Yang, "Sparsity-induced similarity measure and its applications," *IEEE Trans. Circuits Syst. Vedio Technol.*, vol. 26, no. 4, pp. 613–626, Apr. 2016.

[30] S. Gao, I. W.-H. Tsang, and L.-T. Chia, "Laplacian sparse coding, hypergraph Laplacian sparse coding, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 92–104, Jan. 2013.

[31] J. Li and C.-Y. Lu, "A new decision rule for sparse representation based classification for face recognition," *Neurocomputing*, vol. 116, pp. 265–271, Sep. 2013.

[32] J. Gou, Y. Zhan, X. Shen, Q. Mao, and L. Wang, "Two-phase representation based classification," in *Proc. 16th Pacific-Rim Conf. Multimedia*, Gwangju, South Korea, Sep. 2015, pp. 265–274,

[33] L. Qiao, S. Chen, and X. Tan, "Sparsity preserving projections with applications to face recognition," *Pattern Recognit.*, vol. 43, no. 1, pp. 331–341, Jan. 2010.

[34] G. Zhou, Z. Lu, and Y. Peng, "L_1-graph construction using structured sparsity," *Neurocomputing*, vol. 120, pp. 441–452, Nov. 2013.

[35] B. Cheng, J. Yang, S. Yan, Y. Fu, and T. S. Huang, "Learning with L_1-graph for image analysis," *IEEE Trans. Image Process.*, vol. 19, no. 4, pp. 858–866, Apr. 2010.

[36] J. Gou, L. Du, K. Cheng, and Y. Cai, "Discriminative sparsity preserving graph embedding," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, Canada, Jul. 2016, pp. 4250–4257.

[37] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition," in *Proc. ICCV*, Barcelona, Spain, Nov. 2011, pp. 471–478.

[38] Y. Xu, D. Zhang, J. Yang, and J.-Y. Yang, "A two-phase test sample sparse representation method for use with face recognition," *IEEE Trans. Circuits Syst. Vedio Technol.*, vol. 21, no. 9, pp. 1255–1262, Sep. 2011.

[39] S. Zeng, J. Gou, and X. Yang, "Improving sparsity of coefficients for robust sparse and collaborative representation-based image classification," *Neural Comput. Appl.*, to be published, doi: 10.1007/s00521-017-2900-4.

[40] Y. Xu, Q. Zhu, Z. Fan, M. Qiu, Y. Chen, and H. Liu, "Coarse to fine K nearest neighbor classifier," *Pattern Recognit. Lett.*, vol. 34, pp. 980–986, Jul. 2013.

[41] W. Li, Q. Du, F. Zhang, and W. Hu, "Collaborative-representation-based nearest neighbor classifier for hyperspectral imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 389–393, Feb. 2015.

[42] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Roy Statist. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.

[43] G. Gao, L. Zhang, J. Yang, L. Zhang, and D. Zhang, "Reconstruction based finger-knuckle-print verification with score level adaptive binary fusion," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5050–5062, Dec. 2013.

[44] A. Frank and A. Asuncion. (2013). UCI machine learning repository. School of Information and Computer Science. University of California, Irvine, CA, USA, accessed on Apr. 10, 2011. [Online]. Available: http://archive.ics.uci.edu/ml

[45] J. Alcala-Fdez *et al.*, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Multiple-Valued Logic Soft Comput.*, vol. 17, nos. 2–3, pp. 255–287, 2011.
[46] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.
[47] S. Garcia and F. Herrera, "An extension on 'Statistical comparisons of classifiers over multiple data sets' for all pairwise comparisons," *J. Mach. Learn. Res.*, vol. 9, pp. 2677–2694, Dec. 2008.
[48] J. Derrac, S. Garcia, and D. Molinac, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm Evol. Comput.*, vol. 1, pp. 3–18, Mar. 2011.

HONGXING MA is currently pursuing the Ph.D. degree in computer science with Shaanxi Normal University, Xi'an, China. He is currently an Associate Professor with the College of Electrical and Information Engineering, North Minzu University, Yinchuan, China. His current research interests include image processing and pattern recognition.

JIANPING GOU (M'16) received the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2012. He is currently an Associate Professor with the School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang, China. His current research interests include pattern recognition and machine learning.

XILI WANG received the Ph.D. degree in computer science from Xidian University, Xi'an, China, in 2004. She is currently a Professor with the College of Computer Science, Shaanxi Normal University, Xi'an, China. Her current research interests include intelligent information processing, and image perception and understanding.

JIA KE received the Ph.D. degree in computer science from Jiangsu University, Zhenjiang, China, in 2013. She is currently an Associate Professor with the School of Computer Science and Telecommunication Engineering, Jiangsu University. Her current research interests include image processing and pattern recognition.

SHAONING ZENG received the B.S. degree in aircraft power engineering and the M.S. degree in software engineering from Beihang University, Beijing, China, in 2004 and 2007, respectively. Since 2009, he has been a Lecturer with Huizhou University, China. His current research interests include pattern recognition and image recognition.