A Local Mean Representation-based K-Nearest Neighbor Classifier

JIANPING GOU and WENMO QIU, Jiangsu University, Jiangsu Key Laboratory of Security Technology for Industrial Cyberspace, China ZHANG YI, Sichuan University, China YONG XU, Shenzhen Graduate School, Harbin Institute of Technology, China QIRONG MAO and YONGZHAO ZHAN, Jiangsu University, China

K-nearest neighbor classification method (KNN), as one of the top 10 algorithms in data mining, is a very simple and yet effective nonparametric technique for pattern recognition. However, due to the selective sensitiveness of the neighborhood size k, the simple majority vote, and the conventional metric measure, the KNN-based classification performance can be easily degraded, especially in the small training sample size cases. In this article, to further improve the classification performance and overcome the main issues in the KNN-based classification, we propose a local mean representation-based k-nearest neighbor classifier (LM-RKNN). In the LMRKNN, the categorical k-nearest neighbors of a query sample are first chosen to calculate the corresponding categorical k-local mean vectors, and then the query sample is represented by the linear combination of the categorical k-local mean vectors; finally, the class-specific representation-based distances between the query sample and the categorical k-local mean vectors are adopted to determine the class of the query sample. Extensive experiments on many UCI and KEEL datasets and three popular face databases are carried out by comparing LMRKNN to the state-of-art KNN-based methods. The experimental results demonstrate that the proposed LMRKNN outperforms the related competitive KNN-based methods with more robustness and effectiveness.

CCS Concepts: • AI Technology \rightarrow Machine Learning; • Data Mining \rightarrow Classification algorithm;

Additional Key Words and Phrases: K-nearest neighbor classification, local mean vector, representation, pattern recognition

© 2019 Association for Computing Machinery.

2157-6904/2019/04-ART29 \$15.00

https://doi.org/10.1145/3319532

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. U1836220, 61502208, 61672267 and 61672268), the Natural Science Foundation of Jiangsu Province of China (Grant No. BK20150522), International Postdoctoral Exchange Fellowship Program of China Postdoctoral Council (No. 20180051), the Research Foundation for Talented Scholars of JiangSu University (Grant No. 14JDG037) and Open Foundation of Artificial Intelligence Key Laboratory of Sichuan Province (Grant No. 2017RYJ04).

Authors' addresses: J. Gou (corresponding author), W. Qiu, Q. Mao, and Y. Zhan, School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang 212013, PR China; emails: goujianping@ujs.edu.cn, qiuwenmo@163.com, Mao_qr@mail.ujs.edu.cn, yzzhan@ujs.edu.cn; Y. Zhang, School of Computer Science, Sichuan University, Chengdu 610065, Sichuan, PR China; email: zhangyi@scu.edu.cn; Y. Xu, Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, Guangdong, PR China; email: yongxu@ymail.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

ACM Reference format:

Jianping Gou, Wenmo Qiu, Zhang Yi, Yong Xu, Qirong Mao, and Yongzhao Zhan. 2019. A Local Mean Representation-based K-Nearest Neighbor Classifier. ACM Trans. Intell. Syst. Technol. 10, 3, Article 29 (April 2019), 25 pages.

https://doi.org/10.1145/3319532

1 INTRODUCTION

K-nearest neighbor (KNN) classification, since the benchmark KNN method was first introduced in Ref. [1], has been extensively studied and widely applied in the fields of machine learning and pattern recognition [2–6]. Owing to its superiorities of the simple implementation, high effectiveness, and easy intuitiveness, KNN has been viewed as one of the top 10 algorithms in data mining [7]. The basic idea of the conventional KNN is that the query sample belongs to the class with the most frequency among its *k*-nearest neighbors by a simple majority vote. Essentially, as a nonparametric classification technique, these good superiorities of KNN are determined by its attractive properties [8], i.e., only one parameter of the neighborhood size k, the asymptotically optimal performance in Bayes sense under sufficient conditions [1, 9], and no need of *a priori* knowledge of the classification data.

Generally, the KNN-based classification possesses the good performance in many practical applications of pattern recognition. However, the issues related to one parameter k, a simple majority vote, and the conventional metric measure such as Euclidean distance still exist [3, 5, 10]. The issue of sensitivity of choosing k always has a major influence on the performance, especially in the small training sample size cases with existing outliers [11]. How to determine the reasonable value of k plays a very important role in the KNN-based classification. If k is very small, the neighborhood region may contain noisy and imprecise points so as to easily make the wrong classification prediction [10, 12]. In contrast, the neighborhood region with a large value of k may have more existing outliers that heavily degrade the classification performance. Furthermore, the choice of the single and uniform value of k for all testing samples is also unsuitable for classification because each testing sample may come from different classes in order that the KNN-based classification performance could be suffered from such neighborhood choice. To support a suitable value of k for each query sample and overcome the sensitivity of k, some adaptive neighborhood choices for different query samples are proposed in Refs. [6] and [13]-[17]. As we know, most of the KNN-based classifiers employ a simple majority vote that assumes each neighbor has an equal weight for making classification decisions. Obviously, this assumption is not very reasonable in practice. Specifically speaking, different neighbors have different contributions for classification and the reliable neighbors should have more contribution. As a result, each neighbor should be given different weight for making classification decisions and, accordingly, many weighted voting methods for KNN are developed, such as in Refs. [10], [15], and [18]-[21]. Among these weighted KNN methods, the classical one is the distance-weighted k-nearest neighbor classifier (WKNN) [18]. In many KNN-based methods, k-nearest neighbors of each query sample are often sought by using the conventional distance metric measure such as Euclidean distance. Such chosen neighbors may be unreliable and cannot really reflect the similarities and the discrimination of the query samples. Thus, the KNN-based classification performance may significantly depend on the metrics that are adopted to calculate distances between training and testing samples. To address this issue, the weighted distance metrics [22, 23] and the representation-based methods [2, 24–30] are used to compute the similarities of samples to determine k-nearest neighbors of each query sample. Due to the essential discrimination of representation learning [31, 32], the representation coefficients and representation-based distances are often used for choosing nearest neighbors. The locally linear *k*-nearest neighbor method [28] uses representation coefficients to determine the neighbors for visual recognition. The coarse to fine *k*-nearest neighbor classifier (CFKNN) [2] uses the representation-based distances to determine discriminative neighbors of each query sample in two phases for image classification.

In fact, the main issues above can be easily aggravated by the existing outliers, especially in the case of the small training sample size [3, 11, 33]. In other words, the KNN-based classification performance can be heavily degraded by the existing outliers in the neighborhoods. With the aim of improving the performance of KNN and overcoming the negative effect from the existing outliers, many categorical k-nearest neighbor classification methods have been developed in Refs. [3, 5, 8, 33–38]. The classical one of the categorical *k*-nearest neighbor methods is called the local meanbased k-nearest neighbor classifier (LMKNN) using the local mean vector of k-nearest neighbors per class [33]. The LMKNN method is further extended to the kernel version in Ref. [34]. Using the local mean vector, Gou et al. proposed the local mean-based k-nearest centroid neighbor classifier [37] and the local mean-based pseudo-nearest neighbor classifier (LMPNN) [3]. The basic idea of LMPNN is based on both LMKNN and the pseudo-nearest neighbor classifier (PNN) [35]. The extended version of LMPNN is accordingly proposed by using nearest centroid neighborhood in Ref. [36] and the good classification performance of the LMPNN is practically verified in Ref. [39]. Zeng et al. proposed the PNN method that designs the categorical pseudo-nearest neighbor by using the distance-weighted k-nearest neighbors chosen from each class [35]. Due to the advantages of the local mean vectors in the KNN-based classification, it has been well utilized in some related new classification methods [5, 8, 38, 40-42].

To further improve the KNN-based classification performance, we present the proposed local mean representation-based k-nearest neighbor classifier (LMRKNN) method in this article, with the purpose of overcoming the issues above, especially the sensitivity of k with the existing outliers in the small sample size cases. In the LMRKNN, the categorical k-local mean vectors computed by k-nearest neighbors per class are first achieved for reflecting different local sample distributions of each class in order to reduce the sensitivity of the single and uniform value of k for classifying all testing samples. Since different local subclasses in each class could have different importance for classification, each query sample is represented by the linear combination of the categorical k-local mean vectors, with expectations to make the true local mean vector play great importance to represent its own class. Instead of using the simple majority vote, the representation-based distances calculated by the categorical k-local mean vectors are employed for making the classification decision. The classification performance of the LMRKNN is fully explored by carrying out extensive experiments on many real datasets downloaded from the University of California, Irvine (UCI) and Knowledge Extraction based on Evolutionary Learning (KEEL) repositories and several popular face databases, in comparisons with the KNN, WKNN, LMKNN, CFKNN, PNN, and LMPNN methods. The effectiveness and robustness of the proposed LMRKNN is demonstrated by the comparative experimental results. Thus, three key good properties of the LMRKNN method are held as follows:

- The categorical *k*-local mean vectors instead of *k*-nearest neighbors are adopted to capture the geometric and discriminant information of data and to reduce the sensitivity of *k*.
- —Each query sample is represented by the linear combination of the categorical *k*-local mean vectors and each local mean vector is weighted through the representation to provide the different true importance for its own class in the classification stage.
- The class-specific representation-based distances calculated by the categorical *k*-local mean vectors are employed as the classification decision with more pattern discrimination.

In this article, we extend our conference paper, which has been published in the 29th IEEE International Conference on Tools with Artificial Intelligence (ICTAI) [43]. With the exception of

the contents of the conference paper, we still mainly have presented the related work, the detailed description, and analysis of the proposed LMRKNN, the extensive experiments, and the computational complexities. The remainder of this article is organized as follows. Section 2 briefly reviews the related KNN-based classification methods. Section 3 proposes LMRKNN method in detail. Section 4 gives the analysis of the rationale of the proposed LMRKNN. Section 5 presents extensive experiments, and Section 6 gives the discussions about the experimental observations of the LM-RKNN and the computational complexities of the comparative methods. Finally, the conclusions are drawn in Section 7.

2 RELATED METHODS

KNN is one of the simplest and most effective nonparametric techniques and its many outstanding extensions have been introduced recently. In this section, we briefly describe three variants, i.e., LMKNN [33], LMPNN [3], and CFKNN [2] that are related to our work due to using the local mean vector in the LMKNN and LMPNN and representation-based distances in the CFKNN. For the ease of the descriptions in what follows, we first uniformly introduce the notations. In general classification problems, we assume the set of *n* training samples within *M* classes is denoted as $X = [x_1, x_2, \ldots, x_n] = [X_1, X_2, \ldots, X_M] \in \mathbb{R}^{d \times n}$ and the set of all the class labels is $C = \{c_1, c_2, \ldots, c_M\}$. The class label l_i of any one sample x_i is $l_i \in C$, and $X_j = [x_1^j, x_2^j, \ldots, x_{n_j}^j]$ denotes n_j training samples from the class c_j .

2.1 LMKNN

The LMKNN method [33], as a classical extension of KNN, is mainly based on the local mean vector of k-nearest neighbors in each class. It is more robust to the sensitivity of k, especially under the small sample training size situations with the existing outliers. Given a query sample y, its k-nearest neighbors from each class c_j are chosen by using the Euclidean distance metric as

$$d(y, x_i^j) = \sqrt{(y - x_i^j)^T (y - x_i^j)}.$$
 (1)

Let $X_{kj}^{NN}(y) = [x_{1j}^{NN}, x_{2j}^{NN}, \dots, x_{kj}^{NN}]$ be the set of *k*-nearest neighbors from class c_j . Then, the local mean vector of the categorical *k*-nearest neighbors in class c_j is calculated as

$$m_{kj}^{NN} = \frac{1}{k} \sum_{i=1}^{k} x_{ij}^{NN}.$$
 (2)

Accordingly, $d(y, m_{kj}^{NN})$ for each class is computed and the query sample y is finally classified into the class with the minimum distance between each categorical local mean vector and y among all classes.

2.2 LMPNN

The LMPNN method is also a promising extension of KNN that is based on the categorical pseudonearest neighbor of each query sample [3]. The pseudo-nearest neighbor from each class is designed by using weighted k-local mean vectors corresponding to k-nearest neighbors. For a given query sample y, its categorical k-nearest neighbors using Euclidean distance metric are first found from class c_j and denoted as $X_{kj}^{NN}(y) = [x_{1j}^{NN}, x_{2j}^{NN}, \dots, x_{kj}^{NN}]$. Then, the categorical k-local mean vectors in terms of k-nearest neighbors in each class c_j are calculated as

$$\bar{x}_{ij}^{NN} = \frac{1}{i} \sum_{f=1}^{i} x_{fj}^{NN}, i = 1, \dots, k.$$
 (3)

After computing each distance $d(y, \bar{x}_{ij}^{NN})$, we use *k*-local mean vectors to design the categorical pseudo-nearest neighbor x_i^{PN} of *y* as follows:

$$d(y, x_j^{PN}) = \left(1 \times d(y, \bar{x}_{1j}^{NN}) + \frac{1}{2} \times d(y, \bar{x}_{2j}^{NN}) + \dots + \frac{1}{k} \times d(y, \bar{x}_{kj}^{NN})\right), j = 1, 2, \dots, M.$$
(4)

Finally, the query sample y is classified into the class with the closest pseudo-nearest neighbor x_j^{PN} that has the minimum distance $d(y, x_j^{PN})$ among M classes. It should be noted that both LMPNN and LMKNN are the same as the nearest neighbor classifier when k = 1, and the number of k is generally no greater than the smallest number n_j of the training samples from class c_j among all the classes.

2.3 CFKNN

The CFKNN method is a two-phase linear representation-based k-nearest neighbor classifier, which uses the representation-based distances to make classification decisions [2]. In the first phase, the given query sample y is coarsely represented by the linear combination of all the training samples as

$$y = \alpha_1 x_1 + \alpha_2 x_2 + \dots + \alpha_n x_n, \tag{5}$$

where α_i is the representation coefficient of training sample x_i and $\alpha = [\alpha_1 \alpha_2 \cdots \alpha_n]^T$. These representation coefficients in α can be often solved as $\bar{\alpha} = (X^T X + \lambda I)^{-1} X^T y = [\bar{\alpha}_1 \bar{\alpha}_2 \cdots \bar{\alpha}_n]^T$, where I is an identity matrix. The representation contribution of x_i to y is calculated by the representation-based distance as follows:

$$e_i = \|y - x_i \bar{\alpha}_i\|^2. \tag{6}$$

Then, the n_0 training samples corresponding to the first n_0 smallest e_i are chosen as the representative samples and denoted as $Z = [z_1, z_2, ..., z_{n_0}] \in \mathbb{R}^{d \times n_0}$. In the second phase, the query sample y is further finely represented as

$$y = \beta_1 z_1 + \beta_2 z_2 + \dots + \beta_{n_0} z_{n_0}.$$
 (7)

Just like solving α in Equation (5), the optimal $\beta = [\beta_1 \beta_2 \cdots \beta_n]^T$ can be easily achieved as $\bar{\beta} = (Z^T Z + \gamma I)^{-1} Z^T y = [\bar{\beta}_1 \bar{\beta}_2 \cdots \bar{\beta}_{n_0}]^T$. Using the representation-based distance $d_i = ||y - z_i \bar{\beta}_i||^2$ as the similarity metric, the first *k* training samples corresponding to the first *k* smaller d_i are determined as *k*-nearest neighbors of *y*. Finally, the query sample *y* is classified into the class, which is most frequent among *k*-nearest neighbors.

3 THE PROPOSED LMRKNN

In this section, we describe the idea and procedure of the proposed LMRKNN that integrates both the multi-local mean vectors and the representation-based distance. The purpose of the LMRKNN is to improve the KNN-based classification performance and reduce the negative effect from the main issues of KNN, especially in the small sample size cases with the existing outliers.

3.1 The Basic Idea

To overcome the existing outliers in the k-neighborhood, the local mean vector of k-nearest neighbors from each class is calculated for classification, such as in the LMKNN [33], but the sensitivity of k still exists. As we know, to assign a fixed uniform value to k for classifying all query samples is unreasonable, and the value of k should be different for each query sample [16]. Since the local sample distributions in each class are very different and the chosen nearest samples with different numbers for the given query sample can well reflect the different local sample distributions per class in the KNN-based classification, the numbers of chosen categorical nearest samples for the

query sample should be quite different. Furthermore, the local sample distributions of each class for different query samples are also different. Thus, the uniform value of k for all classes and all query samples degrades the KNN-based classification performance. Since the categorical multilocal mean vectors can well represent the different local sample distributions of each class in the region of the k-neighborhood, they can lessen the sensitive degree of k and improve the robustness of the KNN-based classification performance. Therefore, the categorical k-local mean vectors are employed in the classifier design of the proposed LMRKNN.

In the LMRKNN, the categorical k-local mean vectors are involved in making classification decision. Essentially speaking, the different local mean vectors correspond to the different similar local subclasses of each class for the query sample and can reflect more discrimination information. Moreover, the closer local mean vector should have greater importance to represent its own class for classification. As a result, instead of a simple majority vote among k-nearest neighbors as the classification decision, the categorical k-local mean vectors in the LMRKNN are adaptively weighted by using them to represent the query sample before classification. The representation coefficient of each local mean vector is viewed as the weighted contribution to classifying the query sample. As stated in the CFKNN [2], the representation-based distances can truly reflect the similarities of samples. Thus, the class-specific representation-based distances between the query sample and the categorical k-local mean vectors are used as a similarity metric for the final classification decision of the proposed LMRKNN. Thus, the proposed LMRKNN holds the superior properties of KNN, multi-local mean vectors, and representation-based distance, and can overcome the main issues in the KNN-based classification, and its effectiveness and robustness are fully verified in Sections 4 and 5.

3.2 The LMRKNN Algorithm

The LMRKNN classification procedure is described in detail in this section. For the given query sample y, its k-nearest neighbors chosen from each class are first found by using Euclidean distance metric in Equation (1). The set of k-nearest neighbors from class c_j is still indicated as $X_{kj}^{NN}(y) = [x_{1j}^{NN}, x_{2j}^{NN}, \dots, x_{kj}^{NN}]$. To well reflect different local sample distributions in each class and reduce the sensitivity of the uniform value of k for all classes, the categorical k-local mean vectors are computed by using the chosen k-nearest neighbors from each class as follows:

$$m_{ij}^{NN} = \frac{1}{i} \sum_{h=1}^{i} x_{hj}^{NN}, i = 1, \dots, k.$$
(8)

Let $\bar{X}_{j}^{y} = [m_{1j}^{NN}, m_{2j}^{NN}, \dots, m_{kj}^{NN}] \in \mathbb{R}^{d \times k}$ denote the set of *k*-local mean vectors in class c_{j} .

To adaptively get weighted contribution of each categorical local mean vector to representing its own class in the process of classifying the query sample y, we use the categorical k-local mean vectors in each class c_i to linearly represent y in terms of

$$y = s_{1j}m_{1j}^{NN} + s_{2j}m_{2j}^{NN} + \dots + s_{kj}m_{kj}^{NN} = \bar{X}_j^y S_{kj},$$
(9)

where $S_{kj} = [s_{1j}, s_{2j}, \ldots, s_{kj}]^T$, and s_{ij} is the representation coefficient associated with the local mean vector m_{ij}^{NN} . If \bar{X}_j^y is a nonsingular square matrix, the coefficient vector S_{kj} can be easily obtained by

$$\bar{S}_{kj} = \left(\bar{X}_j^y\right)^{-1} y. \tag{10}$$

However, \bar{X}_{i}^{y} is always a non-square matrix. In such case, S_{ki} can be obtained as

$$\bar{S}_{kj} = \left(\left(\bar{X}_j^y \right)^T \bar{X}_j^y \right)^{-1} \left(\bar{X}_j^y \right)^T y, \tag{11}$$

A Local Mean Representation-based K-Nearest Neighbor Classifier

if $(\bar{X}_j^y)^T \bar{X}_j^y$ is not a singular matrix. To overcome the singularity of \bar{X}_j^y or $(\bar{X}_j^y)^T \bar{X}_j^y$, the representation coefficient vector S_{kj} can be solved by constraining the l_2 -norm regularization item of S_{kj} as follows:

$$\min\{\|y - \bar{X}_{j}^{y}S_{kj}\|^{2} + \tau \|S_{kj}\|^{2}\},\tag{12}$$

where τ is a regularized parameter. To solve the optimal S_{kj} , we adopt the Lagrange multiplier to define the following function

$$F(S_{kj}) = \|y - \bar{X}_j^y S_{kj}\|^2 + \tau \|S_{kj}\|^2.$$
(13)

Next, the derivative of $F(S_{kj})$ with respect to S_{kj} can be calculated as

$$\frac{\partial F(S_{kj})}{\partial S_{kj}} = -2\left(\bar{X}_{j}^{y}\right)^{T}\left(y - \bar{X}_{j}^{y}S_{kj}\right) + 2\tau S_{kj}$$

$$= -2\left(\bar{X}_{j}^{y}\right)^{T}y + 2\left(\left(\bar{X}_{j}^{y}\right)^{T}\bar{X}_{j}^{y} + \tau I\right)S_{kj}^{j},$$
(14)

where *I* is a identity matrix. Let $\frac{\partial F(S_{kj})}{\partial S_{kj}} = 0$; we obtain the optimal solution \bar{S}_{kj} of S_{kj}

$$\bar{S}_{kj} = \left(\left(\bar{X}_j^y \right)^T \bar{X}_j^y + \tau I \right)^{-1} \left(\bar{X}_j^y \right)^T y.$$
(15)

It should be noted that the representation coefficient vector S_{kj} in Equation (9) can be easily achieved with an optimal closed solution in terms of Equation (15), and the solution \bar{S}_{kj} is very discriminative for classification and robust to noise samples [44, 45]. Thus, the way of this solution is adopted in the LMRKNN. The optimal coefficient \bar{s}_{ij} in \bar{S}_{kj} is regarded as the adaptive weight that reflects the importance of m_{ij}^{NN} from its own class for representing and classifying y.

Using the categorical representation coefficients in \bar{S}_{kj} , we design the class-specific representation-based distance between the query sample y and each class c_j as

$$d_{kj}(y) = \left\| y - \bar{X}_{j}^{y} \bar{S}_{kj} \right\|^{2}.$$
 (16)

The representation-based distances in Equation (16) are employed as the discriminative classification decision. The good power of the pattern discrimination from Equation (16) stems from the multi-local mean vectors and the different weighted contribution for classification from each local mean vector by representation per class. We will further analyze the benefit of the class-specific representation-based distance in Equation (16) in Section 4. Finally, the query sample y is classified into the class that has the minimum representation-based distance in Equation (17) among all classes.

$$l_y = \arg\min_{c_j} d_{kj}(y), j = 1, 2, \dots, M,$$
 (17)

where l_y is the class label of y. As mentioned above, the pseudo codes of the proposed LMRKNN method are detailedly summarized in Algorithm 1.

3.3 Differences between LMRKNN and LMKNN, LMPNN, CFKNN

In this section, we clearly elaborate the differences between LMRKNN and three related methods, i.e., LMKNN, LMPNN, and CFKNN, in order to further emphasize the motivation why we design LMRKNN.

 LMRKNN vs. LMKNN: LMKNN first chooses k-nearest neighbors for a query sample from each class and then use them to calculate the class-specific local mean vector. It determines the class label of the given query sample by using the distance between the query sample and each class-specific local mean vector. In LMKNN, the single and uniform value of k

ALGORITHM 1: The LMRKNN algorithm

Require

y: a query sample, *k*: the number of nearest neighbors.

X: a training set with *n* training samples.

 X_i : denotes a training subset from the class c_i with n_i training samples.

 $C = \{c_1, c_2, \ldots, c_M\}$: the set of *M* classes.

Ensure:

The class label of the query sample y.

Step 1: Calculate the distance between the query sample y and the training samples from each class c_j .

for i = 1 to n_i **do**

$$d(y, x_i^j) = \sqrt{(y - x_i^j)^T (y - x_i^j)}$$

end for

Step 2: Seek *k*-nearest neighbors from each class c_j corresponding to the training samples determined by the first *k* smaller distances $d(y, x_i^j)$, denoted as $X_{kj}^{NN}(y) = [x_{1j}^{NN}, x_{2j}^{NN}, \dots, x_{kj}^{NN}]$.

Step 3: Calculate the categorical k local mean vectors. Each local mean vector m_{ij}^{NN} is obtained by the average of the first *i*-nearest neighbors in class c_j .

for i = 1 to k **do** $m_{ij}^{NN} = \frac{1}{i} \sum_{h=1}^{i} x_{hj}^{NN}$

end for

Denote $\bar{X}_{j}^{y} = [m_{1j}^{NN}, m_{2j}^{NN}, \dots, m_{kj}^{NN}]$

Step 4: The query sample y is represented by a linear combination of the categorical k-local mean vectors in each class c_i .

$$y = s_{1j}m_{1j}^{NN} + s_{2j}m_{2j}^{NN} + \dots + s_{kj}m_{kj}^{NN} = \bar{X}_{j}^{y}S_{kj}, \ S_{kj} = [s_{1j}, s_{2j}, \dots, s_{kj}]^{T}$$

Step 5: S_{kj} is solved as $\bar{S}_{kj} = ((\bar{X}_{j}^{y})^{T}\bar{X}_{j}^{y} + \tau I)^{-1}(\bar{X}_{j}^{y})^{T}y.$

Step 5: S_{kj} is solved as $S_{kj} = ((X_j) | X_j + \tau I) | (X_j) | y$. **Step 6:** Calculate the categorical representation-based distance between the query sample y and each class.

for j = 1 to M **do** $d_{kj}(y) = ||y - \bar{X}_j^y \bar{S}_{kj}||^2$

end for

Step 7: The query sample *y* is classified into the class with the closest categorical representation-based distance.

 $l_y = \arg\min_{c_j} d_{kj}(y).$

is for all classes, which still brings the sensitivity of k to have the negative effect on the classification performance. In its classification decision (i.e., Equation (20)), each nearest neighbor per class gives the identical weight for classification. However, different neighbors should have different weights. Furthermore, one local mean vector per class can not well reflect the different local sample distributions of its own class. Unlike LMKNN, LM-RKNN computes the categorical multi-local mean vectors using k-nearest neighbors of the given query sample and represents the query sample with a linear combination of all the training samples from each class. Through the linear combination, the representation coefficient of each local mean vector in each class to represent the query sample is obtained

as the adaptive weight for classification. Using the representation coefficient, the class label of the query sample is determined by representation-based distances between the query sample and categorical multi-local mean vectors. Compared to LMKNN, LMRKNN employs categorical multi-local mean vectors to fully reflect the local sample distributions of each class, which can overcome the sensitivity of k from the single and uniform value of k for all classes. Meanwhile, the representation-based distance (i.e., Equation (16) or (18)) as the classification decision of LMRKNN makes different local mean vectors have different weighted contribution for classification.

- (2) LMRKNN vs. LMPNN: Both LMRKNN and LMPNN employ the categorical multi-local mean vectors computed by using *k*-nearest neighbors in each class, but they have different classification decisions. In LMPNN, the distance between the categorical pseudo-nearest neighbor and each query sample is indirectly computed and then used as the classification decision. In the process of the classification decision of LMPNN (i.e., Equation (4)), the identical weight $\frac{1}{i}$ for the *i*-th local mean vector from each class *j* is given on its distance $d(y, m_{ij}^{NN})$. In practice, different categorical multi-local mean vectors should have different weighted contribution for different query sample's classification. Thus, the classification decision of LMPNN can not well reflect the discriminant information and local sample distributions of each class. To overcome the issues in LMPNN, LMRKNN designs the class-specific representation-based distance as the classification decision by representing each query sample as a linear combination of the class-specific multi-local mean vectors. And using the representation-based distance (i.e., Equation (16) or (18)) in LMRKNN, each local mean vector from each class has different weight for classification.
- (3) LMRKNN vs. CFKNN: CFKNN uses two-phase representation to find *k*-nearest neighbors of each query sample. In the first phase, the query sample is represented by a linear combination of all the training samples and the representative nearest training samples using the representation-based distance (i.e., Equation (6)) are chosen. In the second phase, the query sample is further represented by a linear combination of the chosen nearest training samples and its *k*-nearest neighbors are determined. Then, the simple majority vote among the chosen *k*-nearest neighbors are used as the classification decision. Unlike CFKNN, LMRKNN adopts different ways of choosing *k*-nearest neighbors and designing the classification decision. In LMRKNN, *k*-nearest neighbors of each query sample per class are first determined by Euclidean distance, and the categorical *k*-local mean vectors are computed by the corresponding class-specific *k*-nearest neighbors. Accordingly, the query sample is represented by a linear combination of categorical multi-local mean vectors. After the representation of the query sample by multi-local mean vectors, the categorical representation-based distances are obtained and used as the classification of LMRKNN.

4 THE ANALYSIS OF LMRKNN

In this section, we further analyze the good properties of the proposed LMRKNN method from the perspective of the categorical representation-based distance and multi-local mean vectors, and accordingly give the intuitive examples. As described in Section 3, the LMRKNN employs a linear representation of the query sample on the basis of k-local mean vectors to get the representation-based distance as the classification decision in each class, in order to achieve good performance. Here, the importance of representation-based distance and multi-local mean vectors for the favorable classification in the LMRKNN are theoretically and visually analyzed.

First of all, we analyze the good property of the categorical representation-based distance as the classification decision in the LMRKNN, in comparison with the similar classification decision in the LMKNN [33]. As stated in the LMRKNN, the categorical representation-based distance between the query sample and each class in Equation (16) is in fact the distance from the query sample to the sum of the weighted multi-local mean vectors per class as follows:

$$d_{kj}(y) = \|y - \bar{X}_j^y \bar{S}_{kj}\|^2 = \|y - \sum_{i=1}^k \bar{s}_{ij} m_{ij}^{NN}\|^2.$$
(18)

Furthermore, each local mean vector represents a local sample distribution in the class-specific k-neighborhood and should have different appropriate importance of classification for its own class. Through the linear representations of the query sample by using the categorical multi-local mean vectors, the adaptive importance of classification for the query sample from each local mean vector in each class is represented by the representation coefficient. We can use the derivative of the class-specific representation-based distance with respect to each local mean vector as the weighted classification contribution from each local mean vector to calculate the classification decision. The derivative of $d_{kj}(y)$ with respect to m_{ij}^{NN} is computed as follows:

$$\frac{\partial d_{kj}(y)}{\partial m_{ij}^{NN}} = \frac{\partial ||y - \bar{X}_j^y \bar{S}_{kj}||^2}{\partial m_{ij}^{NN}}
= \frac{\partial ||y - \sum_{i=1}^k \bar{s}_{ij} m_{ij}^{NN}||^2}{\partial m_{ij}^{NN}}
= -2 \left(y - \sum_{i=1}^k \bar{s}_{ij} m_{ij}^{NN} \right) \bar{s}_{ij}
= -2 \left(y - \sum_{i=1}^k \frac{\bar{s}_{ij}}{i} \left(\sum_{h=1}^i x_{hj}^{NN} \right) \right) \bar{s}_{ij}
= 2 \left(\bar{X}_j^y \bar{S}_{kj} - y \right) \bar{s}_{ij}.$$
(19)

In terms of Equations (16) and (18), we can explain the good benefits of the classification decision in the proposed LMRKNN from two aspects. One is that each local mean vector give the proper weighted contribution in representation-based distance according to its representation coefficient. The other one is that each neighbor in the different multi-local mean vectors fully has different weighted classification importance for classification. Thus, from Equation (19), the categorical representation-based distances are very favorable for classification. In contrast to LM-RKNN, LMKNN uses the distance between the query sample and the local mean vector of k-nearest neighbors in each class c_i as the classification decision

$$d(y, m_{kj}^{NN}) = \left\| y - m_{kj}^{NN} \right\|^2 = \left\| y - \frac{1}{k} \sum_{i=1}^k x_{ij}^{NN} \right\|^2.$$
(20)

0

Similar to Equation (19), the derivative of Equation (20) represents the weight of each neighbor for classification as follows:



Fig. 1. The example of the classification results of the query sample truly from the class 1 in the two-class classification problem using different classification decisions.

$$\frac{\partial d(y, m_{kj}^{NN})}{\partial x_{ij}^{NN}} = \frac{\partial \|y - m_{kj}^{NN}\|^2}{\partial x_{ij}^{NN}} \\
= \frac{\|y - \frac{1}{k} \sum_{i=1}^k x_{ij}^{NN}\|^2}{\partial x_{ij}^{NN}} \\
= -\frac{2}{k} \left(y - \frac{1}{k} \sum_{i=1}^k x_{ij}^{NN}\right) \\
= \frac{2}{k} \left(m_{kj}^{NN} - y\right).$$
(21)

Obviously, each neighbor in the LMKNN gives the identical weight for the classification decision, which is not suitable for classification.

To well verify the meaningfulness of the representation-based distance in the LMRKNN, we give the examples in one two-classification task to analyze the classification decisions used in both LMKNN and LMRKNN, shown in Figures 1 and 2. In fact, the used two-classification dataset originally has 166 attributes within two classes, each of which contains 100 samples that are randomly chosen from each class in the real "Musk1" dataset, as shown in Table 2. For visualization, the



Fig. 2. The example of the classification results of the query sample truly from class 2 in the two-class classification problem using different classification decisions.

dimensionality of the samples is reduced from 166 to 2 by Fisher Criterion [33]. Suppose that n_j training samples are from class c_j in *C* classes, then the Fisher Criterion F(f) is defined as follows

$$F(f) = \frac{\sum_{j=1}^{C-1} \sum_{i=j+1}^{C} P(c_i) P(c_i) (\mu_{fj} - \mu_{fi})^2}{\sum_{i=1}^{C} P(c_i) \sigma_{fi}^2},$$
(22)

where $P(c_i) = n_i / \sum_{j=1}^C n_j$, and μ_{fi} and σ_{fi}^2 are the average and variance on feature f for class c_i , respectively. By the Fisher Criterion F(f), these attributes in this dataset are listed in the descending order, and the first two features are chosen for visualization in Figures 1 and 2. We select a representative query sample from each class and give its intuitive classification results by LMKNN and LMRKNN when the value of k is preset to be seven.

Using Equations (18) and (20), the classification results of two chosen query samples are schematically displayed in Figures 1 and 2. Note that the query samples are denoted as the red squares, the samples in the pink ellipses in green are their *k*-nearest neighbors from each class, and $m_j = m_{kj}^{NN} = \frac{1}{k} \sum_{i=1}^k x_{ij}^{NN}$ is the local mean vector of their *k*-nearest neighbors in LMKNN and $r_j = \bar{X}_j^y \bar{S}_{kj} = \sum_{i=1}^k \bar{s}_{ij} m_{ij}^{NN}$ is the representation-based vector of their *k*-local mean vectors from class c_j in LMRKNN in the two figures. From the point-of-view of people, the query samples in Figures 1 and 2 are classified into their true classes to a great extent according to the local sample

A Local Mean Representation-based K-Nearest Neighbor Classifier

distributions in the regions of the k-neighborhoods. However, in terms of distances from m_i and r_i to the query samples, the LMKNN mistakenly classifies them, but the proposed LMRKNN correctly classifies them. According to the nearest samples in the k-neighborhoods in Figures 1(a) and 2(a), we can see that two query samples can be easily classified into the wrong classes due to the outliers, which are from the different classes with the query samples. That is, the classification performance of the LMKNN can be influenced by the outliers in terms of the distance between the query sample and the local mean vector of k-nearest neighbors in each class, and its classification decision can not provide the proper weighted contribution from each neighbor for classification. Furthermore, we can clearly observe that the representation-based vector r_i of k-local mean vectors computed by k-nearest neighbors in class c_i can well represent the geometrical and discriminant information from its own whole class, shown in Figure 1(c) combined with Figure 1(a), and Figure 2(c) combined with Figure 2(a). The reason can be from the fact that the categorical multi-local mean vectors in the LMRKNN can reflect the different local sample distributions in the k-neighborhood regions and each local mean vector has different importance to represent its own class through the linear representation of the query using multi-local mean vectors. Thus, we can conclude that the class-specific representation-based distance with more discrimination is very effective for classification.

Secondly, we mainly analyze the superior property of the categorical k-local mean vectors used for the classification decision instead of the categorical k-nearest neighbors in terms of cosine correlation coefficients of different points. It has been argued that multi-local mean vectors can represent different local sample distributions of k-neighborhood that can overcome the sensitivity of k and obtain promising classification [3, 5]. Just like the use of cosine correlation coefficients in Ref. [2], the cosine correlation coefficient between points x_i and x_j is calculated as

$$sim(x_i, x_j) = \frac{x_i^T x_j}{\|x_i\| \|x_j\|}.$$
 (23)

Then, the mean of cosine correlation coefficients among the given n points is calculated as

$$sim = \frac{2}{n(n-1)} \sum_{p=1}^{n-1} \sum_{q=p+1}^{n} sim(x_p, x_q).$$
(24)

It is well-known that for any two points x_i and x_j , the larger $sim(x_i, x_j)$ is, the more similar they are. If $sim(x_i, x_j)$ tends to be 1, they are significantly similar with the same direction. Using Equations (23) and (24), we compute the means of cosine correlation coefficients of *k*-local mean vectors and *k*-nearest neighbors from each class and use them to investigate their effect on classification. The mean of cosine correlation coefficients of *k*-local mean vectors are computed as

$$sim(\bar{X}_{j}^{y}) = \frac{2}{k(k-1)} \sum_{p=1}^{k-1} \sum_{q=p+1}^{k} sim(m_{pj}^{NN}, m_{qj}^{NN}),$$
(25)

where $sim(m_{pj}^{NN}, m_{qj}^{NN}) = \frac{(m_{pj}^{NN})^T m_{qj}^{NN}}{\|m_{pj}^{NN}\| \|m_{qj}^{NN}\|}$. And the mean of cosine correlation coefficients of *k*-nearest neighbors are computed as

$$sim(X_{kj}^{NN}(y)) = \frac{2}{k(k-1)} \sum_{p=1}^{k-1} \sum_{q=p+1}^{k} sim(x_{pj}^{NN}, x_{qj}^{NN}),$$
(26)

where $sim(x_{pj}^{NN}, x_{qj}^{NN}) = \frac{(x_{pj}^{NN})^T x_{qj}^{NN}}{\|\|x_{pj}^{NN}\|\|\|x_{qj}^{NN}\|\|}$. According to Equation (25), if *k*-local mean vectors of the query samples in each class are more similar to each other, their means of cosine correlation co-



Fig. 3. The means of the cosine correlation coefficients of LMRKNN and LMKNN at different values of k for each class on Wine ("KN-i" and "KM-i" represent k-nearest neighbors and k-local mean vectors in each class c_i , respectively).



Fig. 4. The classification errors of LMRKNN and LMKNN with varying different values of k for each class on Wine ("LMKNN-i" and "LMRKNN-i" represent LMRKNN and LMKNN in each class c_i , respectively).

efficients will be very large. Similarly, the very similar *k*-nearest neighbors make their means of cosine correlation coefficients to be too large in terms of Equation (26). In other words, the values of the means of cosine correlation coefficients in each class reflect the degrees to represent their own class, and the larger means of cosine correlation coefficients should enhance the query samples to be more correctly classified into its own classes.

Using Equations (25) and (26), the means of cosine correlation coefficients of k-local mean vectors and k-nearest neighbors from each class are achieved on real Wine dataset with three classes and displayed in Figures 3 and 4. The description of Wine is shown in Table 2. It should be noted that both local mean vectors and nearest neighbors used in each mean of cosine correlation coefficients and their corresponding query samples are from the same class. From Figure 3, we can observe that the means of cosine correlation coefficients of k-local mean vectors in each class are significantly larger than the corresponding ones of *k*-nearest neighbors. This fact implies that the classification results of the LMRKNN using k-local mean vectors are very better than the ones of the LMKNN using k-nearest neighbors, which are verified in Figure 4. Furthermore, it can be seen in Figure 3 that the means of cosine correlation coefficients in one class are often larger than the ones in the other classes, and accordingly, its classification performance in the class is also better than the ones in the other classes. Such fact is also well reflected in Figure 4. For example, the classification results in class 1 are often best among three classes, and the classification results in class 3 are often better than the ones in class 2 among the values of k. To further demonstrate this point, the minimum and average classification results in three classes among the range of k are illustrated in Table 1. Also, we can see that the means of cosine correlation coefficients of multilocal mean vectors in each class at first slowly increase and then tend to be stable with the increase of k in Figure 3; accordingly, the classification errors of the LMRKNN in each class often quickly

	Mii	nimum err	ors	Average errors			
The method	Class 1	Class 2	Class 3	Class 1	Class 2	Class 3	
LMKNN	5.08	28.81	22.03	10.17	30.51	33.90	
LMRKNN	1.69	15.25	3.39	6.09	28.36	9.49	

Table 1. The Classification Errors (%) of LMKNN and LMRKNN in Each Class on Wine among All Values of k

degrade and the lower classification errors are always obtained at larger values of k in Figure 4. This indicates that the proposed LMRKNN can use more nearest neighbors for the good classification. The better classification performance with the larger means of cosine correlation coefficients of k-local mean vectors in Figures 3 and 4 confidently reveals that the multi-local mean vectors can well provide different suitable local sample distributions that have the more discriminant similarities with the query samples for favorable classification. Thus, it can be concluded that k-local mean vectors computed by k-nearest neighbors in the proposed LMRKNN are more helpful for classification than k-nearest neighbors.

Through the analyses above, it can be found that the superior classification performance of the proposed LMRKNN stems from the multi-local mean vectors and the representation-based distance that become the main properties of the LMRKNN method.

5 EXPERIMENTS

In order to verify the classification performance of the proposed LMRKNN method, we carry out the extensive experiments on many datasets by comparing LMRKNN with the state-of-art KNN-based classifiers such as KNN, WKNN, LMKNN, CFKNN, PNN, and LMPNN in this section.

5.1 Datasets

The summarized information of all the used datasets in the experiments is briefly described in this section. We use two types of datasets including 24 numerical datasets and 3 face image datasets. These real numerical datasets are downloaded from the UCI machine learning repository [46] and the KEEL Repository [47, 48]. The numbers of total samples, testing samples, attributes, and classes of each dataset are listed in Table 2. Among these numerical datasets, "Libras," "Bank," "Cardio," "Image," "Opt," "Landsat," and "Pen" are the abbreviated names of "LibrasMovement," "Banknote," "Cardiotocography," "Image segmentation," "Optdigits," "LandsatSatellite," and "PenDigits," respectively. And most of the numerical datasets are the multi-class classification tasks. In the experiments, each numerical dataset is randomly divided into the sets of training and testing samples 10 times. The final classification results of each method are the averages of the classification errors among 10 runs. Note that the numbers of all samples in most of the chosen real numerical datasets are mostly very small in order to demonstrate the classification performance of the proposed method in the small training sample size cases.

The face image datasets used in the experiments are three popular face databases: Olivetti Research Ltd (ORL), Georgia Tech (GT), and Pose, Illumination, and Expression (PIE). ORL (www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html) contains 40 subjects, each of which has 10 gray scale image samples. All image samples in each subject are collected by varying the lighting, facial expressions, or facial details at different times, and the size of each image sample is 112 × 92 pixels. GT (www.anefian.com/research/face_reco.htm) contains 50 subjects, each of which has 15 frontal and/or tilted facial images collected by the variations in illumination conditions, facial expression, and appearance. Each sample is a color JPEG image and the size of each image is 640 × 480 pixels. PIE (www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html)

Data	Samples	Attributes	Classes	Testing samples
Wine	178	13	3	48
Seeds	210	7	3	45
Balance	625	4	3	135
Vowel	528	10	11	132
CNAE	1,080	856	9	450
Libras	360	90	15	90
Sonar	208	60	2	62
Musk1	476	166	2	130
Bank	1,372	4	2	950
Vehicle	846	18	4	360
Bands	365	19	2	50
Ecoli	307	7	4	104
Yeast	1,299	8	4	635
Iris	150	4	3	45
Wpbc	198	32	3	60
Cardio	2,126	21	10	670
Spambse	4,597	57	2	1,400
Texture	5,500	40	11	1,650
Image	2,310	19	7	770
Opt	5,620	64	10	3,000
Musk2	6,598	166	2	2,400
Landsat	6,435	36	6	2,170
Pen	10,992	16	10	4,000
Letter	20,000	16	26	6,500

Table 2. The Real Numerical Datasets Used in the Experiments



Fig. 5. The examples of the image samples from one subject on each face dataset.

contains 41,368 image samples taken from 68 subjects. The image samples are captured by varying poses, illuminations, and expressions. A subset of PIE from 68 subjects, each of which has 24 samples, is adopted in the experiments. On three face datasets, all original images are aligned and cropped, and then the size of each cropped image sample is resized to 32×32 with 256 gray levels per pixel. Figure 5 shows the image samples of one subject in ORL, GT, and PIE, respectively.

5.2 Experiments on the numerical datasets

In this section, the classification performance of the proposed LMRKNN is highlighted on 24 real numerical datasets in terms of the classification error, compared to KNN, WKNN, LMKNN, CFKNN, PNN, and LMPNN. The experiments are conducted by the 10-trials holdout validation,

A Local Mean Representation-based K-Nearest Neighbor Classifier



Fig. 6. The classification errors of each method via *k* on each numerical dataset.

and the final classification performance evaluations are the averages of the classification errors 10 times with 95% confidence.

From the point-of-view of the sensitivity of the neighborhood size k, the classification results of all competing methods are first compared by varying the values of the parameter k. In the experiments, the range of the neighborhood size k is preset from 1 to 15 with a step of 1. The classification errors of each method on each real numerical dataset at different values of k are displayed in Figures 6 and 7. Note that the differences of the classification errors between CFKNN and the other methods are very significant on the Bank dataset; the classification errors of CFKNN are not illustrated in Figure 6(h). And also, the possible reason of the poor classification results of CFKNN in Figures 6 and 7 is that CFKNN could be often suitable for image classification [2] and not good on the numerical datasets [3]. As can be seen from the experimental results in Figures 6 and 7, the proposed LMRKNN always performs best among the seven methods with different values of k. This clearly reveals LMRKNN is less sensitive to k than the other methods with better classification results. Moreover, the improvements of the classification performance between LMRKNN and the other six methods are often significant at relatively large values of k in most cases. This experimental fact implies that the proposed LMRKNN method can use more nearest neighbors for better classification than the other methods. In addition, as shown in Figures 6 and 7, the curves of the classification errors of the LMRKNN with varying the values of k are always smoother and flatter than other methods on most of the datasets, especially when the value of k is large. Thus, the experimental results via the changes of k show that the proposed LMRKNN holds the less sensitiveness of k with the significant improvement of the classification performance, compared to the competing KNN-based classifiers.



Fig. 7. The classification errors of each method via *k* on each numerical dataset.

To further highlight the effectiveness of the proposed LMRKNN, the best classification results of each competing method among the range of k from 1 to 15 are explored. The minimum classification errors of each method with the corresponding standard deviations (stds) and the optimal values of k in the parentheses on all real numerical datasets are listed in Table 3. Note that the best classification performance among the seven KNN-based methods on each dataset is indicated in bold-face. As shown in Table 3, we can obviously see that the LMRKNN method nearly outperforms the other methods on all datasets. Most importantly, we can find two facts from the experimental results in Table 3. The one is that the proposed LMRKNN method often significantly performs better than the other methods under the situations of the small training sample size problems, such as on Wine, Musk1, CNAE, Libras, and Bands. The other one is that the LMRKNN always achieves the best classification at larger values of k than KNN, WKNN, LMKNN, and PNN, which is also embodied in Figures 6 and 7. Such experimental facts imply that the proposed LMRKNN can obtain good classification performance by fully using more nearest neighbors with the robustness to kand performs well in the small training sample size cases. It should be noted that the LMPNN uses more nearest neighbors for good classification just like LMRKNN. This possible reason is that both LMRKNN and LMPNN are based on multi-local mean vectors. Besides, the average minimum classification error of LMRKNN on all datasets is significantly lower than the other methods, which reveals that the classification performance of LMRKNN is very robust and stable.

The experimental results above are obtained by using ten-trials holdout validation with 95% confidence. However, the comparisons of the state-of-art KNN-based methods may not be statistically convincible. In order to further validate whether the proposed LMRKNN outperforms the other competing methods or not, one of the famous non-parametric statistical tests called the Friedman test [49–51] is performed. The Friedman test first ranks the competing methods on each dataset in terms of the classification errors in Table 3. Specifically speaking, the first method with the best

Data	KNN	WKNN	LMKNN	CFKNN	PNN	LMPNN	LMRKNN
Wine	26.67 ± 4.37	26.25 ± 4.83	24.58 ± 3.90	16.46 ± 6.01	25.00 ± 6.51	23.75 ± 4.93	5.83±2.91
	(1)	(6)	(4)	(15)	(3)	(15)	(15)
Musk1	13.92 ± 3.02	12.85 ± 2.46	9.62 ± 2.06	21.00 ± 2.12	13.31 ± 2.35	10.23 ± 2.27	8.69±1.81
	(4)	(6)	(6)	(12)	(3)	(14)	(14)
Vowel	3.56 ± 1.39	3.56 ± 1.39	3.56 ± 1.39	19.32 ± 2.93	3.56 ± 1.39	3.41 ± 1.57	2.65 ± 1.09
	(1)	(1)	(1)	(1)	(1)	(3)	(8)
Balance	6.15 ± 1.35	4.81 ± 1.27	4.89 ± 1.49	17.56 ± 1.88	4.15 ± 1.27	3.85 ± 1.20	3.11±0.84
	(4)	(7)	(3)	(15)	(3)	(4)	(13)
Seeds	6.89 ± 2.66	6.89 ± 2.66	5.33 ± 3.18	7.11 ± 2.73	5.78 ± 1.87	5.33 ± 2.81	5.11± 2.78
	(1)	(1)	(7)	(15)	(6)	(6)	(3)
CNAE	16.78 ± 1.06	13.60 ± 1.86	9.11±1.33	11.29 ± 1.70	12.80 ± 1.61	10.31 ± 1.47	7.76±1.06
	(3)	(15)	(9)	(6)	(11)	(15)	(15)
Libras	17.33 ± 4.45	16.33 ± 4.06	14.00 ± 4.60	23.44 ± 4.76	16.56 ± 4.58	13.78 ± 4.75	10.78 ± 2.62
	(1)	(5)	(2)	(4)	(2)	(3)	(14)
Bank	0.37 ± 0.20	0.37 ± 0.20	0.33 ± 0.14	9.04 ± 1.33	$0.37 {\pm} 0.20$	0.31 ± 0.14	0.20 ± 0.14
	(1)	(1)	(5)	(15)	(1)	(15)	(8)
Sonar	16.61 ± 4.30	16.13 ± 3.65	15.32 ± 4.11	13.55 ± 4.11	16.61 ± 4.30	14.52 ± 4.02	12.42 ± 2.16
	(1)	(4)	(3)	(4)	(1)	(13)	(12)
Cardio	30.51 ± 1.50	29.22 ± 1.28	29.25 ± 1.26	48.54 ± 2.32	28.75 ± 1.28	26.42 ± 1.13	26.52 ± 1.48
	(1)	(7)	(2)	(15)	(4)	(12)	(15)
Vehicle	33.39 ± 1.37	32.17 ± 1.09	29.33 ± 1.34	29.19 ± 1.67	32.14 ± 1.13	29.64 ± 1.37	28.89 ± 1.13
	(3)	(5)	(3)	(15)	(4)	(9)	(5)
Ecoli	12.00 ± 2.47	11.81 ± 2.21	11.05 ± 1.86	14.29 ± 2.29	12.48 ± 2.31	11.62 ± 1.67	11.05 ± 2.02
	(11)	(12)	(12)	(13)	(9)	(12)	(13)
Bands	35.40 ± 6.19	34.20 ± 7.08	34.80 ± 4.44	34.00 ± 6.67	35.40 ± 6.19	35.20 ± 6.48	29.80 ± 5.20
	(1)	(4)	(2)	(1)	(1)	(2)	(6)
Yeast	39.69 ± 1.51	39.95 ± 1.45	41.17 ± 1.24	44.25 ± 2.38	39.50 ± 1.21	41.70 ± 1.10	38.71±1.14
	(15)	(15)	(14)	(14)	(15)	(15)	(15)
Iris	1.56 ± 1.83	1.11±1.57	1.56 ± 1.50	10.44 ± 4.69	1.11±1.57	1.78 ± 1.41	1.11±1.57
	(6)	(9)	(2)	(10)	(13)	(4)	(11)
Wpbc	26.90 ± 3.65	27.41 ± 5.10	23.97 ± 2.87	24.14 ± 1.63	26.38 ± 5.39	25.17 ± 3.47	23.79 ± 2.12
	(8)	(7)	(15)	(13)	(4)	(15)	(14)
Spambase	19.51 ± 1.22	18.67 ± 1.06	17.44 ± 0.88	40.16 ± 1.12	18.04 ± 1.01	16.54 ± 0.82	11.30 ± 0.73
	(1)	(10)	(11)	(15)	(5)	(15)	(15)
Texture	1.10 ± 0.14	1.02 ± 0.26	0.87 ± 0.16	$1.54{\pm}0.14$	1.02 ± 0.24	0.76 ± 0.12	0.36±0.15
	(1)	(6)	(2)	(10)	(3)	(4)	(7)
Image	7.30 ± 0.83	7.27 ± 0.53	6.86 ± 0.44	14.99 ± 1.13	7.17 ± 0.65	6.39 ± 0.87	6.00±0.36
	(1)	(7)	(4)	(15)	(2)	(8)	(11)
Opt	1.64 ± 0.19	1.51 ± 0.19	1.11 ± 0.21	2.19 ± 0.12	1.46 ± 0.15	1.10 ± 0.17	1.01 ± 0.12
	(1)	(7)	(9)	(15)	(2)	(15)	(15)
Musk2	4.28 ± 0.38	3.92 ± 0.20	3.37 ± 0.22	3.03±0.09	3.79 ± 0.25	3.52 ± 0.18	3.38 ± 0.25
	(4)	(7)	(4)	(1)	(10)	(12)	(5)
Landsat	18.76 ± 0.51	18.18 ± 0.74	15.37 ± 0.45	42.66 ± 0.90	18.14 ± 0.44	15.64 ± 0.85	15.23 ± 0.66
	(7)	(13)	(13)	(15)	(11)	(15)	(15)
Pen	0.70 ± 0.12	0.62 ± 0.07	0.52 ± 0.04	3.77 ± 0.46	0.62 ± 0.11	0.47 ± 0.09	0.45 ± 0.05
	(4)	(8)	(3)	(15)	(4)	(15)	(10)
Letter	4.57 ± 0.28	4.25 ± 0.25	3.77 ± 0.17	32.56 ± 0.50	4.19 ± 0.25	3.28 ± 0.15	3.23±0.17
	(4)	(9)	(4)	(15)	(4)	(15)	(15)
Average	14.40 ± 1.87	13.84 ± 1.89	12.80 ± 1.64	20.19 ± 2.24	13.68 ± 1.93	12.70 ± 1.68	10.72 ± 1.36

Table 3. The Minimum Classification Error Results (%) of Each Method on All the Real Numerical Datasets

Table 4. The Average Ranks of the Competing Methods Using Friedman Test on All Real Numerical Datasets

Method	KNN	WKNN	LMKNN	CFKNN	PNN	LMPNN	LMRKNN
Average rank(R_i)	5.75	4.64	3.39	5.50	4.50	3.08	1.14

performance among all the methods gains the rank of 1, the second one the rank of 2, and so on. When some methods have the same classification performance, the average ranks are assigned for them in such a tie. Let R_i^j be the rank of the *i*th one of *t* methods on the *j*th one of *n* datasets. The average rank of the *i*th method on *n* datasets is $R_i = \frac{1}{n} \sum_{j=1}^n R_i^j$. Under the null hypothesis of the Friedman test, all competing methods nearly achieve a similar classification performance, and the ranks R_i could be identical. The Friedman statistics is defined as

$$\mathcal{X}_{F}^{2} = \frac{12n}{t(t+1)} \left[\sum_{i} R_{i}^{2} - \frac{t(t+1)^{2}}{4} \right].$$
 (27)

Note that when n > 10 and t > 5, Friedman statistics is distributed by X_F^2 with t - 1 degrees of freedom.

In terms of the classification errors in Table 3, we use the Friedman test to investigate the classification performance of the proposed LMRKNN, compared to the other six classifiers. The average rank of each method is listed in Table 4 according to the classification results from Table 3. Using the average ranks, $X_F^2 = 85.162$ is computed by Equation (27). Under the null hypothesis, the mean rank $\overline{R} = 4.0014$ can be achieved if all the competing methods perform similarly and X_F^2 with 7 – 1 degrees of freedom at $\alpha = 0.05$ is $(X_F^2)_{0.05} = 12.592$. As shown in Table 4, the average ranks between the proposed LMRKNN and the other methods are very significant. Moreover, X_F^2 for all competing methods is significantly larger than $(X_F^2)_{0.05}$. Consequently, the non-parametric statistical test demonstrates that seven competing KNN-based classifiers are very different, and the proposed LMRKNN performs significantly.

5.3 Experiments on the image datasets

In this section, the classification performance of the proposed LMRKNN is studied in the highdimensional face datasets including PIE, ORL, and GT in terms of the classification error, compared to KNN, WKNN, LMKNN, CFKNN, PNN, and LMPNN. As we know, the KNN-based classification performance can be easily degraded in the case of the curse of dimensionality. In the experiments, the dimensionality of each image sample is 1024. And we employ ten-trials holdout validation to conduct the experiments on each image dataset, and the averages of the classification errors on 10 trials with 95% confidence are obtained as the final classification results. That is to say, each image dataset is randomly divided into training samples and testing samples 10 times. On each dataset, *l* samples of each class are chosen as the training samples and the others are the testing ones.

The classification performance of the competing KNN-based methods is explored by varying the neighborhood size k in terms of the classification error on each image dataset. The values of k are varied from 1 to l with a step of 1. The classification results of each method via k are shown in Figure 8. We can see that the proposed LMRKNN almost performs best among the competing methods with different values of k, and the improvements of LMRKNN are very significant when k is large. So that the robustness of LMRKNN to the sensitivity of k can be obviously verified. Moreover, the classification errors of LMRKNN at first quickly descend at small values of k and



Fig. 8. The classification errors of each method via k on PIE, ORL, and GT image datasets.

Table 5. The Minimum Classification Errors (%) of Each Method with the Corresponding StandardDeviations (stds) and the Optimal Values of k in the Parentheses on the Image Datasets

Data	l	KNN	WKNN	LMKNN	CFKNN	PNN	LMPNN	LMRKNN
PIE	10	23.61 ± 1.68	23.61 ± 1.68	23.61 ± 1.68	$8.95 {\pm} 0.64$	23.61 ± 1.68	22.71 ± 2.02	7.88±0.77
		(1)	(1)	(1)	(3)	(1)	(2)	(10)
	15	13.50 ± 2.07	13.50 ± 2.07	13.50 ± 2.07	10.03 ± 1.43	13.50 ± 2.07	12.81 ± 2.21	6.31±1.50
		(1)	(1)	(1)	(1)	(1)	(2)	(12)
ODI	5	$11.60 {\pm} 2.07$	$11.60 {\pm} 2.07$	10.50 ± 1.66	$9.80{\pm}2.61$	11.60 ± 2.07	10.70 ± 1.79	7.90±1.47
		(1)	(1)	(2)	(4)	(1)	(4)	(5)
OKL	7	8.00 ± 3.47	7.83 ± 3.47	6.17±3.75	5.33 ± 1.92	8.00 ± 3.47	5.83 ± 3.12	4.00±2.24
		(1)	(4)	(2)	(5)	(1)	(5)	(6)
GT	8	30.51 ± 1.80	29.60 ± 1.76	25.37 ± 2.39	29.71±1.28	29.26 ± 1.77	25.20 ± 2.46	23.20 ± 2.20
		(1)	(6)	(3)	(5)	(3)	(7)	(7)
	10	26.00 ± 2.06	25.28±1.31	22.00±0.63	25.60 ± 3.01	25.36 ± 1.73	21.20 ± 1.44	18.48±1.80
	10	(1)	(6)	(3)	(8)	(2)	(6)	(8)

then tend to be stable when k becomes large. However, unlike LMRKNN, the classification errors of the other methods nearly ascend with the increase of k. In addition, the proposed LMRKNN always performs better at larger values of k than the other methods. This means LMRKNN can use more nearest neighbors for better classification.

The best classification performance of each method among the ranges of k on three image datasets are displayed in Table 5. Note that the best classification result on each dataset among the competing methods is denoted in bold-face. From the experimental results in Table 5, we can clearly observe that the proposed LMRKNN significantly outperforms the other six methods, and the best performance of LMRKNN is obtained at larger values of k than the other methods. Hence,

the proposed method is more effective and robust than KNN, WKNN, LMKNN, CFKNN, PNN, and LMPNN with better classification results.

6 DISCUSSIONS

In this section, we first briefly give the observations from the experiments in Section 5 and then the analysis of the computational complexities of the competitive methods.

6.1 Experimental Observations

From all the comparative experiments on both the real numerical datasets and the image datasets above, we can briefly achieve several observations for the LMRKNN as follows:

- It can reduce the sensitivity of choosing the neighborhood size k with the promising classification performance through multi-local mean vectors of k-nearest neighbors from each class.
- It can generally achieve the better classification performance by using the categorical multilocal mean vectors of more nearest neighbors.
- -It can often perform better in the case of the small training sample size and the curse of dimensionality.
- It is more effective and robust with a good classification performance by adopting the class-specific representation-based distances as the classification decision.

These experimental observations for LMRKNN coincide with the good properties of the categorical multi-local mean vectors and representation-based distance described and analyzed in Sections 3 and 4. Therefore, it has been proved that the proposed LMRKNN is a promising method with effectiveness and robustness for the KNN-based classification in pattern recognition.

6.2 Computational Complexities

To further embody the effectiveness of the proposed LMRKNN, we analyze the comparative efficiency of the LMRKNN, KNN, WKNN, LMKNN, CFKNN, PNN, and LMPNN methods by providing their computational complexities. In the computational complexities of the competing methods, several notations commonly used are the training sample size n, dimensionality of each sample d, and the number k of neighbors. As stated in Ref. [52], the main computational time of classifying the query sample in KNN and WKNN is to search k-nearest neighbors by calculating Euclidean distances from the query sample to all the training samples. Accordingly, the main computational complexities of KNN and WKNN are O(nd + nk + k) and O(nd + nk + 2k), respectively. In LMKNN, the query sample is classified by finding categorical k-local mean vectors, which are computed by the corresponding categorical k-nearest neighbors chosen from the class-specific training samples. So the main computational complexity of LMKNN is O(nd + nk + ckd + cd). In PNN, the query sample is classified by finding the categorical pseudo-nearest neighbors, which are determined by searching categorical k-nearest neighbors and assigning the weight to each distance between the query sample and each neighbor. As a result, the main computational complexity of PNN is O(nd + nk + 2ck). In LMPNN, the query sample is also classified by finding the categorical pseudo-nearest neighbors on the basis of categorical k-local mean vectors, which are calculated by categorical k-nearest neighbors. Thus, the main computational complexity of LMPNN is O(nd + nk + 2ckd + 3ck). In CFKNN, the query sample is classified by mainly using two-phase representation of the query sample. In the first phase, the query sample is represented by a linear combination of all the training samples, and then n_0 representative training samples are chosen to represent the query sample in the second phase; finally, k-nearest neighbors chosen from n_0 representative samples are determined. So the main computational complexity of CFKNN

Data	KNN	WKNN	LMKNN	CFKNN	PNN	LMPNN	LMRKNN
Seeds	0.0056	0.0068	0.0102	0.0505	0.0084	0.0105	0.0132
(210, 45, 7, 3)	0.0050	0.0000	0.0102				
Musk1	0.0545	0.0557	0.0811	0 2710	0.0770	0.0842	0.0010
(476, 130, 166, 2)	0.0345	0.0337	0.0011	0.3719	0.0770	0.0042	0.0910
Libras	0.0205	0.0224	0.0768	0 2060	0.0642	0.0014	0 1/27
(360, 90, 90, 15)	0.0205	0.0224	0.0708	0.2000	0.0042	0.0914	0.1427
Vowel	0.0127	0.0150	0.0510	0 2124	0.0448	0.0506	0.0026
(528,132,10,11)	0.0127	0.0130	0.0319	0.3134	0.0440	0.0390	0.0930
Cardio	0 1667	0 1745	0.4401	8 0250	0 2085	0.4750	0.6420
(2126,670,21,10)	0.1007	0.1/45	0.4491	0.0239	0.3703	0.4730	0.0430
CNAE	3 6024	3 6108	5 7014	6 701	5 6 2 3 8	5 88/1	6 1133
(1080, 450, 865, 9)	5.0024	5.0108	5./014	0.701	J.0238	3.0041	0.1155

Table 6. The Runtime (s) of Each Method on Seed, Musk1, Libras, Vowel, Cardio, and CNAE (the Values in the Parentheses Are the Numbers of Training Samples, Testing Samples, Attributes, and Classes, Respectively)

is $O(n^3 + n_0^3 + n_0n + kn_0)$. In the proposed LMRKNN, the categorical *k*-nearest neighbors and the corresponding *k*-local mean vectors are first obtained, and then the query sample is represented by a linear combination of *k*-local mean vectors per class; finally, the query sample is classified by categorical representation-based distances among all classes. Hence, the main computational complexity of LMRKNN is $O(nd + nk + 2ckd + ck^3 + cd)$.

Combined with the computational complexities of the comparative methods above, we take the Seed, Musk1, Libras, Vowel, Cardio, and CNAE datasets as examples to give the runtime of each method when k = 5. The numbers of training samples, testing samples, attributes, and classes in these datasets are different, as shown in Table 2. Note that all our experiments are carried out in the 64-bit Windows 7 operating system using Intel(R) Core(TM) i7-3770 CPU @3.40GHz 3.40GHz and 12GB of memory. Table 6 shows the runtime of all comparative methods to classify the testing samples on these five datasets. As displayed in Table 6, it is clear that the values of the runtime of all methods are in accordance with their corresponding computational complexities.

7 CONCLUSIONS

This article proposes a local mean representation-based k-nearest neighbor classifier (LMRKNN). The purpose of LMRKNN is to improve the KNN-based classification performance and to overcome the main issues that exist in KNN. In the LMRKNN, the categorical multi-local mean vectors computed by k nearest neighbors in each class are used for truly reflecting the different local sample distributions, which can not only reduce the sensitivity of the neighborhood size k, especially in the small training sample size cases, but also capture more geometrical and discriminant information for classification. To enhance the true importance of each local mean vector for representing its own class in making classification decisions, the query sample is represented by a linear combination of the categorical multi-local mean vectors in each class and then the solved representation coefficient of each local mean vector as the weighted contribution to representing the query sample is used for designing the representation-based distance. This categorical representation-based distance, as the elaborate distance metric, is the similarity between the query sample and the categorical multi-local mean vectors in each class and is used as the classification decision. The representation-based distance can overcome the negative effect of the simple majority vote and the conventional similarity metric in KNN. These good

properties of the proposed LMRKNN for the favorable classification are analyzed in details. The extensive experiments on many real numerical datasets and three image datasets are conducted by comparing LMRKNN with the state-of-art KNN-based methods. The experimental classification results show that the proposed LMRKNN significantly outperforms the competing methods, and it is an effective and robust classifier in pattern recognition.

REFERENCES

- Thomas Cover and Peter Hart. 1967. Nearest neighbor pattern classification. IEEE Trans. Inf. Theory. 13, 1 (1967), 21–27.
- [2] Yong Xu, Qi Zhu, Zizhu Fan, Minna Qiu, Yan Chen, and Hong Liu. 2013. Coarse to fine k nearest neighbor classifier. Pattern Recognit. Lett. 34, 9 (2013), 980–986.
- [3] Jianping Gou, Yongzhao Zhan, Yunbo Rao, Xiangjun Shen, Xiaoming Wang, and Wu He. 2014. Improved pseudo nearest neighbor classification. *Knowledge-Based Syst.* 70 (2014), 361–375.
- [4] Zhiwen Yu, Hantao Chen, Jiming Liuxs, Jane You, Hareton Leung, and Guoqiang Han. 2016. Hybrid k-nearest neighbor classifier. IEEE Trans. Cybern. 46, 6 (2016), 1263–1275.
- [5] Zhibin Pan, Yidi Wang, and Weiping Ku. 2017. A new k-harmonic nearest neighbor classifier based on the multi-local means. *Expert Syst. Appl.* 67 (2017), 115–125.
- [6] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Ruili Wang. 2018. Efficient kNN classification with different numbers of nearest neighbors. *IEEE Trans. Neural Networks Learn. Syst.* 29, 5 (2018), 1774–1785.
- [7] Xindong Wu, Vipin Kumar, J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, and Zhihua Zhou. 2008. Top 10 algorithms in data mining. *Knowl. Inf. Syst.* 14, 1 (2008), 1–37.
- [8] Boyu Li, Yun Wen Chen, and Yan Qiu Chen. 2008. The nearest neighbor algorithm of local probability centers. *IEEE Trans. Syst. Man Cybern.* 38, 1 (2008), 141–154.
- [9] T. Wagner. 1971. Covergence of the nearest neighbor rule. IEEE Trans. Inf. Theory. 17, 5 (1971), 566-571.
- [10] Jianping Gou, Lan Du, Yuhong Zhang, and Taisong Xiong. 2012. A new distance-weighted k-nearest neighbor classifier. J. Inf. Comput. Sci. 9, 6 (2012) 1429–1436.
- [11] Keinosuke Fukunaga. 2013. Introduction to statistical pattern recognition. Academic Press, San Diego, CA.
- [12] Zhun-Ga Liu, Quan Pan, and Jean Dezert. A new belief-based k-nearest neighbor classification method. Pattern Recognit. 46, 3 (2013), 834–844.
- [13] Jigang Wang, Predrag Neskovic, and Leon N. Cooper. 2006. Neighborhood size selection in the k-nearest-neighbor rule using statistical confidence. *Pattern Recognit.* 39, 3 (2006), 417–423.
- [14] Nicols Garcia-Pedrajas, Juan A. Romero del Castillo, and Gonzalo Cerruela-Garcia. 2017. A proposal for local k values for k-nearest neighbor rule. *IEEE Trans. Neural Networks Learn. Syst.* 28, 2 (2017), 470–475.
- [15] Ankit Vidyarthi, and Namita Mittal. 2016. AVNM: A voting based novel mathematical rule for image classification. Comput. Methods Programs Biomed. 137 (2016) 195–201.
- [16] Faruk Bulut and Mehmet Fatih Amasyali. 2017. Locally adaptive k parameter selection for nearest neighbor classifier: One nearest cluster. Pattern Anal. Appl. 20, 2 (2017), 415–425.
- [17] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Debo Cheng. 2017. Learning k for kNN Classification. ACM Trans. Intell. Syst. Technol. 8, 3 (2017), 43.
- [18] Sahibsingh A. Dudani. 1976. The distance-weighted k-nearest neighbor rule. IEEE Trans. Syst. Man Cybern. 4 (1976), 325–327.
- [19] Jianping Gou, Lan Du, and Taisong 2012. Xiong, Weighted k-nearest centroid neighbor classification. J. Comput. Inf. Syst. 8 (2012), 851–860.
- [20] Ping Li, Jianping Gou, and Hebiao Yang. 2017. The distance-weighted k-nearest centroid neighbor classification. J. Intell. Inf. Hiding Multimedia Signal Process. 8, 3 (2017), 611–622.
- [21] Daniel Mateos-Garc, Jorge Garca-Gutirrez, and JosC. Riquelme-Santos. 2016. An evolutionary voting for k-nearest neighbours. Expert Syst. Appl. 43 (2016), 9–14.
- [22] Kilian Q. Weinberger, and Lawrence K. Saul. 2009. Distance metric learning for large margin nearest neighbor classification. J. Mach. Learn. Res. 10 (2009), 207–244.
- [23] Roberto Paredes and Enrique Vidal. 2006. Learning weighted metrics to minimize nearest-neighbor classification error. IEEE Trans. Pattern Anal. Mach. Intell. 28, 7 (2006), 1100–1110.
- [24] Jian Zhang and Jian Yang. 2014. Linear reconstruction measure steered nearest neighbor classification framework. Pattern Recognit. 47, 4 (2014), 1709–1720.
- [25] Wei Li, Qian Du, Fan Zhang, and Wei Hu. 2015. Collaborative-representation-based nearest neighbor classifier for hyperspectral imagery. *IEEE Geosci. Remote Sens.* 12, 2 (2015), 389–393.
- [26] Nan Zhang, Jian Yang, and Jian-Jun Qian. 2012. Component-based global k-NN classifier for small sample size problems. Pattern Recognit. Lett. 33, 13 (2012), 1689–1694.

A Local Mean Representation-based K-Nearest Neighbor Classifier

- [27] Xiaozhao Fang, Yuwu Lu, Zhengming Li, Lei Yu, and Yan Chen. 2014. Kernel representation-based nearest neighbor classifier. Opt. 125, 10 (2014), 2320–2326.
- [28] Qingfeng Liu and Chengjun Liu. 2017. A novel locally linear KNN method with applications to visual recognition. IEEE Trans. Neural Networks Learn. 28, 9 (2017), 2010–2021.
- [29] Hongxing Ma, Jianping Gou, Xili Wang, Jia Ke, and Shaoning Zeng. 2017. Sparse Coefficient-Based k-Nearest Neighbor Classification. IEEE Access 5 (2017) 16618–16634.
- [30] Si-Bao Chen, Yu-Lan Xu, Chris H. Q. Ding, and Bin Luo. 2018. A nonnegative locally linear KNN model for image recognition. *Pattern Recognit.* 83 (2018), 78–90.
- [31] Zechao Li, Jing Liu, Jinhui Tang, and Hanqing Lu. 2015. Robust structured subspace learning for data representation. IEEE Trans. Pattern Anal. Mach. Intell. 37, 10 (2015), 2085–2098.
- [32] Zechao Li, Jinhui Tang, and Tao Mei. 2018. Deep collaborative embedding for social image understanding. IEEE Trans. Pattern Anal. Mach. Intell. DOI: 10.1109/TPAMI.2018.2852750
- [33] Yoshihiro Mitani and Yoshihiko Hamamoto. 2006. A local mean-based nonparametric classifier. Pattern Recognit. Lett. 27, 10 (2006), 1151–1159.
- [34] Seiji Hotta, Kiyasu Kiyasu, and Sueharu Miyahara. 2004. Pattern recognition using average patterns of categorical k-nearest neighbors. In Proceedings of the 17th IEEE International Conference on Pattern Recognition (ICIP'04), 412–415.
- [35] Yong Zeng, Yupu Yang, and Liang Zhao. 2009. Pseudo nearest neighbor rule for pattern classification. Expert Syst. Appl. 36, 2 (2009), 3587–3595.
- [36] Hongxing Ma, Xili Wang, and Jianping Gou. 2016. Pseudo nearest centroid neighbor classification. In Frontier Computing, Springer Singapore (2016), 103–115.
- [37] Jianping Gou, Yi Zhang, Lan Du, and Taisong Xiong. 2012. A local mean-based k-Nearest centroid neighbor classifier. Comput. J. 55, 9 (2012), 1058–1071.
- [38] Yong Zeng, Yupu Yang, and Liang Zhao. 2009. Nonparametric classification based on local mean and class statistics. Expert. Syst. Appl. 36, 4 (2009), 8443–8448.
- [39] Shengyun Liang, Yunkun Ning, Huiqi Li, Lei Wang, Zhanyong Mei, Yingnan Ma, and Guoru Zhao. 2015. Feature selection and predictors of falls with foot force sensors using KNN-based algorithms. *Sens.* 15, 11 (2015), 29393–29407.
- [40] Jian Yang, Lei Zhang, Jing-yu. Yang, and David Zhang. 2010. From classifiers to discriminators: A nearest neighbor rule induced discriminant analysis. *Pattern Recognit.* 44, 7 (2010), 1387–1402.
- [41] Jing Chai, Hongwei Liu, Bo Chen, and Zheng Bao. 2010. Large margin nearest local mean classifier. Signal Process. 90, 1 (2010), 236–248.
- [42] Jianping Gou, Hongxing Ma, Wei Ou, Shaoning Zeng, Yunbo Rao, and Hebiao Yang. 2019. A generalized mean distance-based k-nearest neighbor classifier. *Expert Syst. Appl.* 115 (2019), 356–372.
- [43] Jianping Gou, Wenmo Qiu, Qirong Mao, Yongzhao Zhan, Xiangjun Shen, and Yunbo Rao. 2017. A multi-local means based nearest neighbor classifier. In Proceedings of the 29th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'17), Boston, Massachusetts, Nov. 6–8, (2017), 448–452.
- [44] Qi Zhu, Daoqiang Zhang, Han Sun, and Zhengming Li. 2015. Combining L1-norm and L2-norm based sparse representations for face recognition. Int. J. Light Electron Opt. 126, 7–8 (2015), 719–724.
- [45] Jianping Gou, Yong Xu, David Zhang, Qirong Mao, Lan Du, and Yongzhao Zhan. 2018. Two-phase linear reconstruction measure-based classification for face recognition. *Inf. Sci.* 433 (2018), 17–36.
- [46] Moshe Lichman. 2017. UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. Retrieved March 18, 2017 from http://archive.ics.uci.edu/ml/index.php.
- [47] Knowledge Extraction based on Evolutionary Learning. Retrieved March 18, 2017 from http://www.keel.es/.
- [48] Jess Alcala-Fdez, Alberto Fernandez, Julin Luengo, Joaqun Derrac, Salvador Garcia, Luciano Sanchez, and Francisco Herrera. 2011. KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. J. Mult.-Valued Log. S. 17, 2–3 (2011), 255–287.
- [49] Janez Demsar. 2006. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 7 (2006), 1-30.
- [50] Salvador Garcia and Francisco Herrera. 2008. An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons. J. Mach. Learn. Res. 9 (2008), 2677–2694.
- [51] Joaqun Derrac, Salvador Garcia, Daniel Molina, and Francisco Herrera. 2011. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* 1, 1 (2011), 3–18.
- [52] Wangmeng Zuo, David Zhang, and Kuanquan Wang. 2008. On kernel difference-weighted k-nearest neighbor classification. Pattern Anal. Appl. 11, 3–4 (2008), 247–257.

Received August 2018; revised March 2019; accepted March 2019