Expert Systems With Applications 115 (2019) 356–372

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A generalized mean distance-based k-nearest neighbor classifier

Jianping Gou[a,*], Hongxing Ma[b], Weihua Ou[c], Shaoning Zeng[d], Yunbo Rao[e], Hebiao Yang[a]

[a] *School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang, Jiangsu, 212013, PR China*
[b] *College of Electrical and Information Engineering, North Minzu University, Yinchuan, Ningxia, 750021, PR China*
[c] *School of Big Data and Computer Science, Guizhou Normal University, Guiyang, Guizhou, 550025, PR China*
[d] *School of Information Science and Technology, Huizhou University, Huizhou, Guangdong, 516007, PR China*
[e] *School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan, 610054, PR China*

ARTICLE INFO

Article history:
Received 3 February 2018
Revised 11 August 2018
Accepted 11 August 2018
Available online 13 August 2018

Keywords:
K-nearest neighbor rule
Local mean vector
Generalized mean distance
Pattern recognition

ABSTRACT

K-nearest neighbor (KNN) rule is a well-known non-parametric classifier that is widely used in pattern recognition. However, the sensitivity of the neighborhood size k always seriously degrades the KNN-based classification performance, especially in the case of the small sample size with the existing outliers. To overcome this issue, in this article we propose a generalized mean distance-based k-nearest neighbor classifier (GMDKNN) by introducing multi-generalized mean distances and the nested generalized mean distance that are based on the characteristic of the generalized mean. In the proposed method, multi-local mean vectors of the given query sample in each class are calculated by adopting its class-specific k nearest neighbors. Using the achieved k local mean vectors per class, the corresponding k generalized mean distances are calculated and then used to design the categorical nested generalized mean distance. In the classification phase, the categorical nested generalized mean distance is used as the classification decision rule and the query sample is classified into the class with the minimum nested generalized mean distance among all the classes. Extensive experiments on the UCI and KEEL data sets, synthetic data sets, the KEEL noise data sets and the UCR time series data sets are conducted by comparing the proposed method to the state-of-art KNN-based methods. The experimental results demonstrate that the proposed GMDKNN performs better and has the less sensitiveness to k. Thus, our proposed GMDKNN with the robust and effective classification performance could be a promising method for pattern recognition in some expert and intelligence systems.

1. Introduction

Since the standard *K*-nearest neighbor (KNN) rule was first introduced for classification (Cover & Hart, 1967), its study has drawn much more attention by many researchers in pattern recognition and machine learning. Because of the simplicity, effectiveness and implementation of the KNN-based classification, KNN has been viewed as one of the top 10 algorithms in data mining (Wu, Kumar, Quinlan, & Ghosh, 2008). Thus, the KNN-based classification methods could be easily applied in many practical classification tasks in the field of expert and intelligence systems. Generally speaking, most of the variants of KNN efficiently determine the class label of the query samples by using both only one parameter k of the neighborhood size and the simple majority vote among k nearest neighbors. But such classification decision can be easily influenced by the sensitivity of k and it can also aggravate the sensitivity of k, so as to heavily degrade the KNN-based classification performance especially in the small sample size cases with the existing outliers.

As we know, the sensitivity of k is a key issue in the KNN-based classification (Gou, Du, Zhang, & Xiong, 2012; Wu et al., 2008). So far, some kinds of ways such as the adaptive choices of k, the weighted voting methods and distance metric learning are always employed to overcome this problem with the purpose of improving the classification performance. Firstly, with regard to the adaptive choices of k for different testing samples, the strategies that the values of k are adaptively determined for different query samples were proposed in the literatures (Bulut & Amasyali, 2017; Garcia-Pedrajas, Castillo, & Cerruela-Garcia, 2017; Zhang, Li, Zong, Zhu, & Cheng, 2017; Zhang, Li, Zong, Zhu, & Wang, 2017; Zhong, Guo, Gao, Shan, & Zheng, 2017). Secondly, the simple majority vote among k nearest neighbors with the identical voting weight for each neighbor often aggravates the degree of the sensitive issue. Generally,

[*] Corresponding author.
E-mail addresses: goujianping@ujs.edu.cn (J. Gou), mhx_nx@163.com (H. Ma), ouweihuahust@gmail.com (W. Ou), zsn@hzu.edu.cn (S. Zeng), cloudrao@gmail.com (Y. Rao), yhbjj@ujs.edu.cn (H. Yang).

https://doi.org/10.1016/j.eswa.2018.08.021
0957-4174/© 2018 Elsevier Ltd. All rights reserved.

different samples in the region of k-neighborhood should have different contributions to classification. So many weight voting methods for k nearest neighbors were proposed, such as in the literatures (Bicego & Loog, 2016; Bo, Lu, & Wang, 2017; Dudani, 1976; Gou et al., 2012; Gou, Xiong, & Kuang, 2011; Jung & Lee, 2017; Li, Gou, & Yang, 2017; Vidyarthi & Mittal, 2016). Among the them, the first famous weighted voting for KNN is the distance-weighted k-nearest neighbor rule (WKNN) (Dudani, 1976). And also, the weighted k-nearest neighbor rule was revisited from the perspective of a classifier combining and its classification improvements have been demonstrated (Bicego & Loog, 2016). Thirdly, using Euclidean distance to measure the similarities between k nearest neighbors and the query samples could make the sensitive issue obvious. For solving this problem, distance metric learning (Domeniconi & Gunopulos, 2002; Weinberger & Saul, 2009) was proposed to properly choose nearest neighbors and determine the similarity metrics, and also some representation-based KNN methods were successfully proposed recently (Fang, Lu, Li, Yu, & Chen, 2014; Li, Du, Zhang, & Hu, 2015; Ma, Gou, Wang, Ke, & Zeng, 2017; Xu et al., 2013; Zhang & Yang, 2014; Zhang, Yang, & Qian, 2012).

Furthermore, from the point of view of the sensitivity of k, the existing outliers in the region of k-neighborhood can seriously aggravate the sensitive issue and heavily damage the KNN-based classification. If k is small, the outliers can dominate the classification decision. Otherwise, the k-neighborhood may contain more outliers. To reduce the negative influence on the KNN-based classification performance from the existing outliers, especially in the small sample size cases, Mitani and Hamamoto proposed the local mean-based k-nearest neighbor classifier (LMKNN) (Mitani & Hamamoto, 2006) and Zeng et al. the pseudo nearest neighbor classifier (PNN) (Zeng, Yang, & Zhao, 2009). The LMKNN first calculates the distance between each query sample and the local mean vector of k-nearest neighbors from each class, and then classifies the query sample into the class with the minimum distance among all the classes. The PNN uses the sum of weighted distances between each query sample and k nearest neighbors from each class to design the pseudo nearest neighbor and classifies the query sample into the class with the closest pseudo nearest neighbor among all classes. Based on the idea of LMKNN using local mean vectors of k-nearest neighbors for all classes, there are many new KNN-based methods such as in the literatures (Gou, Yi, Du, & Xiong, 2012; Gou et al., 2014; Li, Chen, & Chen, 2008; Pan, Wang, & Ku, 2017; Tran, Le, Vinh, Nguyen, & Nguyen, 2017; Yang, Zhang, Yang, & Zhang, 2011; Zeng, Yang, & Zhao, 2009) to further overcome the existing outliers and improve the classification performance. Among these extensions, Gou et al. proposed the local mean-based pseudo nearest neighbor classifier (LMPNN) (Gou et al., 2014), Tran et al. the weighted local mean-based k-nearest neighbor classifier (Tran et al., 2017) and Pan et al. the multi-local means-based k-harmonic nearest neighbor classifier (MLMKHNN) (Pan et al., 2017). On the basis of the LMKNN and PNN, LMPNN first employs k local mean vectors calculated by k nearest neighbors from each class, and then designs the pseudo nearest neighbor by computing the sum of weighted distances between the query sample and k local mean vectors per class, finally classifies the query sample into the class with the closest pseudo nearest neighbor among all the classes. In the MLMKHNN, k local mean vectors calculated by k nearest neighbors from each class are used to design the harmonic mean distance per class, and the query sample is classified into the class with the minimum harmonic mean distance among all the classes. These classification methods above, which mainly use local mean vectors of nearest neighbors, are more robust to the existing outliers in the region of k-neighborhood with the less sensitiveness to k and the satisfactory performance.

To further overcome the sensitivity of k and improve the KNN-based classification performance, we propose a generalized mean distance-based k-nearest neighbor method (GMDKNN). As argued in LMPNN (Gou et al., 2014) and MLMKHNN (Pan et al., 2017), the multi-local mean vectors of k nearest neighbors in each class can represent the different local sample distributions of its own class, and avoid the single and unique value of k from which the sensitivity of k easily results. As such, in the proposed GMDKNN k nearest neighbors of the query sample are first found from each class and then used to calculate the corresponding k local mean vectors. And then, due to the superior fact that the harmonic mean distance can reflect the different weighted contributions from different multi-local mean vectors for classification (Pan et al., 2017), we further extend the harmonic mean distance to its general form that is called generalized distance mean in the GMDKNN (Luukka, 2009; 2010; Oh & Kwak, 2016). Since generalized distance mean can differently handle data samples, especially when a data includes the outliers (Oh & Kwak, 2016), it can always give more appropriate different contributions from multi-local mean vectors to classification than harmonic mean distance. In the proposed method, k generalized mean distances are calculated in each class by using k local mean vectors of k nearest neighbors per class. The multi-generalized mean distances can fully utilize each local mean vector for good classification. Finally, we design a nested generalized mean distance in each class by using the categorical multi-generalized mean distances to further reduce the sensitive issue of k. The nested generalized mean distance per class is employed as the classification decision rule in the GMDKNN, and the query sample is assigned into the class with the minimum nested generalized mean distance among all the classes. To verify the proposed method, we compare it to the state-of-art KNN-based methods on many data sets. Extensive experiments have shown that the proposed GMDKNN method performs well with more robustness to k and the comparative classification performance.

In this article, the work is a full extension of our conference paper, which has been published in the IEEE 2017 International Conference on Security, Pattern Analysis, and Cybernetics (Ma et al., 2017). The main innovations of the work in this article are embodied in the following two aspects:

(a) Multi-generalized mean distances based on multi-local mean vectors in each class are designed in the proposed GMDKNN. They can well reflect the different classification contributions from the different local mean vectors. Moreover, the different classification contributions from the same local mean vector can be provided in different multi-generalized mean distances.
(b) The nested generalized mean distance as the classification decision rule in the proposed GMDKNN is designed by using multi-generalized mean distances in each class. It can fully employ multi-local mean vectors and well reduce the sensitivity of the neighborhood size k.

The rest of the article is organized as follows. In Section 2, we briefly review the related KNN-based methods using local mean vectors. In Section 3, we propose the GMDKNN method and give its analysis in details. In Section 4, we comprehensively present comparative experiments on many data sets. Finally, we give the conclusions in Section 5.

2. The related KNN-based classifiers

In this section, we briefly review three related KNN-based classifiers, which use local mean vectors of k nearest neighbors per class for each testing sample to design different classification decisions. They are the local mean-based k-nearest neighbor classifier (LMKNN) (Mitani & Hamamoto, 2006), the local mean-based pseudo nearest neighbor classifier (LMPNN) (Gou et al., 2014) and the multi-local means-based k-harmonic nearest neighbor classifier (MLMKHNN) (Pan et al., 2017). Moreover, we give the merits and demerits of these three methods in order to emphasize their properties. For the representation convenience in the following sections, we first give out some common notations. In pattern classification, a training set with n training samples from m classes in d-dimensional feature space is denoted as $X = \{x_i \in \mathbb{R}^d\}_{i=1}^{n}$, m classes are represented as a set $C = \{\omega_1, \omega_2, \dots, \omega_m\}$, and the class label of x_i is $c_i \in C$. The training subset from j-th class is recorded as $X^j = \{x_{ij} \in \mathbb{R}^d\}_{i=1}^{n_j}$.

2.1. LMKNN

LMKNN is a simple and effective algorithm to overcome the existing outliers in the KNN-based classification. Given a query or testing sample $y \in \mathbb{R}^d$ in the LMKNN, its k nearest neighbors from each class are first found by using Euclidean distance as

$$d(y, x_{ij}) = \sqrt{(y - x_{ij})^T (y - x_{ij})}\ . \tag{1}$$

Let $X_j^{NN} = \{x_{ij}^{NN} \in \mathbb{R}^d\}_{i=1}^{k}$ be the set of k nearest neighbors of y from class ω_j. Then, the local mean vector of k nearest neighbors per class is calculated as

$$u_{kj}^{NN} = \frac{1}{k} \sum_{i=1}^{k} x_{ij}^{NN}\ . \tag{2}$$

Finally, the query sample y is classified into the class that has the minimum Euclidean distance between the local mean vector u_{kj}^{NN} and y among all the classes in Eq. (3).

$$\omega = \arg\min_{\omega_j} d(y, u_{kj}^{NN}),\ j = 1, 2, \dots, m\ . \tag{3}$$

Based on the rationale of LMKNN, it has been proven that it could overcome the negative influence on the KNN-based classification from the existing outliers (Mitani & Hamamoto, 2006). However, since LMKNN has the single value of the neighborhood size k per class and uniform value of k for all classes (Pan et al., 2017), and also each neighbor has the same weight contribution to classification, it can still easily get the misclassification (Gou et al., 2014).

2.2. LMPNN

LMPNN is a very successful extension of both LMKNN (Mitani & Hamamoto, 2006) and PNN (Zeng et al., 2009). It has more robustness to the sensitivity of k and can well overcome the existing outliers especially in the small sample size cases.

Given a query sample y, LMPNN first finds k nearest neighbors of y from each class using Euclidean distance in Eq. (1) and the chosen k nearest neighbors from class ω_j are also indicated as the set $X_j^{NN} = \{x_{ij}^{NN} \in \mathbb{R}^d\}_{i=1}^{k}$. Note that k nearest neighbors are sorted in an ascending order in terms of their corresponding Euclidean distances to y. Then, k local mean vectors are computed by using k nearest neighbors in each class. Let $U_j^{NN} = \{u_{ij}^{NN} \in \mathbb{R}^d\}_{i=1}^{k}$ denote the k local mean vectors from class ω_j, where u_{ij}^{NN} is defined as

$$u_{ij}^{NN} = \frac{1}{i} \sum_{l=1}^{i} x_{lj}^{NN},\ 1 \le i \le k\ . \tag{4}$$

The corresponding Euclidean distances between k local mean vectors and y are denoted by $d(y, u_{1j}^{NN}), d(y, u_{2j}^{NN}), \cdots, d(y, u_{kj}^{NN})$. And the weights for k local mean vectors per class are assigned, and the weight w_i^j for the local mean vector u_{ij}^{NN} from the class ω_j is defined as:

$$w_i^j = \frac{1}{i},\ i = 1, \dots, k\ . \tag{5}$$

Finally, the distance between the pseudo nearest neighbor x_j^{PNN} and y in class ω_j is calculated as

$$\begin{aligned} d(y, x_j^{PNN}) = \big(&w_1^j \times d(y, u_{1j}^{NN}) + w_2^j \times d(y, u_{2j}^{NN}) \\ &+ \dots + w_k^j \times d(y, u_{kj}^{NN})\big)\ . \end{aligned} \tag{6}$$

The query sample y is classified into the class that has the closest local mean-based pseudo nearest neighbor according to Eq. (6) among all the classes as

$$\omega = \arg\min_{\omega_j} d(y, x_j^{PNN}),\ j = 1, 2, \dots, m\ . \tag{7}$$

As stated in the LMPNN, its good classification performance benefits from the multi-local mean vectors of k-nearest neighbors and pseudo nearest neighbor based on the multi-local mean vectors for each class (Gou et al., 2014). However, the uniform weight mechanism for all classes cannot fully use the discrimination information of nearest samples for classification.

2.3. MLMKHNN

MLMKHNN is a new extension of KNN that employs the harmonic mean distance as the classification decision rule on the basis of multi-local mean vectors of k nearest neighbors from each class for each query sample.

In the MLMKHNN, k nearest neighbors of the given query sample y from each class in terms of Euclidean distance are first found and the corresponding k local mean vectors based on k nearest neighbors are determined by the same way as in the LMPNN. The k nearest neighbors of y, the corresponding k local mean vectors and the Euclidean distances between the corresponding k local mean vectors and y in class ω_j are denoted as $X_j^{NN} = \{x_{ij}^{NN} \in \mathbb{R}^d\}_{i=1}^{k}$, $U_j^{NN} = \{u_{ij}^{NN} \in \mathbb{R}^d\}_{i=1}^{k}$ and $D_j^{NN} = \{d(y, u_{1j}^{NN}), d(y, u_{2j}^{NN}), \cdots, d(y, u_{kj}^{NN})\}$, respectively. Then, the harmonic mean distance of k local mean vectors to y per class is calculated as

$$H(y, U_j^{NN}) = \frac{k}{\sum_{i=1}^{k} \frac{1}{d(y, u_{ij}^{NN})}},\ j = 1, 2, \dots, m\ . \tag{8}$$

Finally, the query sample y is assigned into the class that has the minimum harmonic mean distance among all the classes according to Eq. (9).

$$\omega = \arg\min_{\omega_j} H(y, U_j^{NN}),\ j = 1, 2, \dots, m\ . \tag{9}$$

Note that LMKNN, LMPNN and MLMKHNN can degrade into KNN when $k = 1$, and they have the same classification performance.

In terms of the rationale of MLMKHNN, its good classification performance benefits from the multi-local mean vectors and the harmonic mean distance based on the multi-local mean vectors for each class (Pan et al., 2017). However, the harmonic mean distance cannot well differently handle data samples and properly give classification importance from data samples (Luukka, 2009; 2010; Oh & Kwak, 2016).

3. The proposed GMDKNN

In this section, we first give the preliminary of the proposed method that is entitled the generalized mean distance-based k-nearest neighbor algorithm (GMDKNN). The preliminary of the GMDKNN is the generalized mean distance, which is used for designing GMDKNN. And then we describe the proposed GMDKNN in detail, with the purpose of further improving the KNN-based classification performance and overcoming the sensitiveness of the neighborhood size k.

3.1. Preliminary of GMDKNN

In the proposed GMDKNN, the generalized mean distance as its preliminary plays an important role in classification. We first give the concept of the generalized mean and then discuss the generalized mean distance in details with the aim of elaborating the rationale behind the usage of generalized mean distance in the GMDKNN.

Given a set of k positive real number $\{z_1, z_2, \ldots, z_k\}$, the generalized mean is define as

$$G_p(z_1, z_2, \ldots, z_k) = (\frac{1}{k}\sum_{i=1}^{k} z_i^p)^{\frac{1}{p}} \ , \tag{10}$$

where p is a non-zero real number. In the KNN-based classification, the Euclidean distances between the query sample y and its k nearest neighbors $\{x_i^{NN}\}_{i=1}^k$ are denoted as the set $\{d(y, x_i^{NN})\}_{i=1}^k$. Using Eq. (10), the generalized mean distance of $\{d(y, x_i^{NN})\}_{i=1}^k$ (i.e. the generalized mean of these distances) is defined as

$$G(y, \{x_i^{NN}\}_{i=1}^k) = \left(\frac{1}{k}\sum_{i=1}^{k}\left(d(y, x_i^{NN})\right)^p\right)^{\frac{1}{p}} . \tag{11}$$

The generalized mean distance in Eq. (11) is to measure the distance similarity between a query sample and its k nearest neighbors. As argued in the work Oh and Kwak (2016), the generalized mean distance of the set $\{d(y, x_i^{NN})\}_{i=1}^k$ can be simply expressed by the linear combination of the distances in this set as follows:

$$\sum_{i=1}^{k}\left(d(y, x_i^{NN})\right)^p = \{w_i \times d(y, x_i^{NN})\}_{i=1}^k \ , \tag{12}$$

where $w_i = \left(d(y, x_i^{NN})\right)^{p-1}$. Eq. (12) means that the generalized mean distance is more influenced by the smaller distance when $p < 1$, and the degree of the influence increases as p descends. Thus, the generalized mean distance can strengthen the classification contribution from the neighbors with very small distances and well overcome the negative influence from the existing outliers in the region of k-neighborhood.

Furthermore, from the point view of the weighted classification contributions of neighbors, the rationale behind the usage of the generalized mean distance in the proposed GMDKNN can be highlighted by comparing the generalized mean distance to the harmonic mean distance and the arithmetic mean distance (Pan et al., 2017). Note that the harmonic mean distance is special case of the generalized mean distance when $p = -1$ and the arithmetic mean distance is another special case when $p = 1$. Obviously, in terms of Eq. (11) the harmonic mean distance of the set $\{d(y, x_i^{NN})\}_{i=1}^k$ is defined as

$$H(y, \{x_i^{NN}\}_{i=1}^k) = \frac{k}{\sum_{i=1}^{k}\frac{1}{d(y, x_i^{NN})}} \ . \tag{13}$$

And also the arithmetic mean distance of the set $\{d(y, x_i^{NN})\}_{i=1}^k$ is easily defined as

$$A(y, \{x_i^{NN}\}_{i=1}^k) = \frac{\sum_{i=1}^{k} d(y, x_i^{NN})}{k} \ . \tag{14}$$

As stated in the MLMKHNN (Pan et al., 2017), $\frac{\partial A(y, \{x_i^{NN}\}_{i=1}^k)}{\partial d(y, x_i^{NN})}$ represents the weight of the classification contribution from ith nearest neighbor x_i^{NN}.

$$\frac{\partial A(y, \{x_i^{NN}\}_{i=1}^k)}{\partial d(y, x_i^{NN})} = \frac{\partial\left(\frac{\sum_{i=1}^{k} d(y, x_i^{NN})}{k}\right)}{\partial d(y, x_i^{NN})} = \frac{1}{k} \ . \tag{15}$$

Eq. (15) means that $d(y, x_i^{NN})$ of each neighbor x_i^{NN} has identical classification contribution through the arithmetic mean distance. However, the weight of the classification contribution from ith nearest neighbor x_i^{NN} $\frac{\partial H(y, \{x_i^{NN}\}_{i=1}^k)}{\partial d(y, x_i^{NN})}$ is different in the harmonic mean distance in terms of Eq. (16).

$$\begin{aligned} \frac{\partial H(y, \{x_i^{NN}\}_{i=1}^k)}{\partial d(y, x_i^{NN})} &= \frac{\partial\left(\frac{k}{\sum_{i=1}^{k}\frac{1}{d(y, x_i^{NN})}}\right)}{\partial d(y, x_i^{NN})} \\ &= k \times \left(\frac{1}{d(y, x_i^{NN}) \times \sum_{i=1}^{k}\frac{1}{d(y, x_i^{NN})}}\right)^2 \\ &= \frac{1}{k} \times \frac{H^2(y, \{x_i^{NN}\}_{i=1}^k)}{d^2(y, x_i^{NN})} \ . \end{aligned} \tag{16}$$

In contrast to the arithmetic mean distance, the harmonic mean distance has more weight contribution to classification from the nearest sample with a closer distance to the testing sample y. In other words, the harmonic mean distance is more available for classification than the arithmetic mean distance in pattern recognition. For the generalized mean distance, the weight of the classification contribution from ith nearest neighbor x_i^{NN} is represented as

$$\begin{aligned} \frac{\partial G(y, \{x_i^{NN}\}_{i=1}^k)}{\partial d(y, x_i^{NN})} &= \frac{\partial\left(\left(\frac{1}{k}\sum_{i=1}^{k}\left(d(y, x_i^{NN})\right)^p\right)^{\frac{1}{p}}\right)}{\partial d(y, x_i^{NN})} \\ &= \frac{1}{k} \times \left(\frac{1}{k}\sum_{i=1}^{k}\left(d(y, x_i^{NN})\right)^p\right)^{\frac{1-p}{p}} \left(d(y, x_i^{NN})\right)^{(p-1)} \\ &= \frac{1}{k} \times \left(G(y, \{x_i^{NN}\}_{i=1}^k)\right)^{(1-p)} \left(d(y, x_i^{NN})\right)^{(p-1)} \\ &= \frac{1}{k} \times \left(\frac{G(y, \{x_i^{NN}\}_{i=1}^k)}{d(y, x_i^{NN})}\right)^{(1-p)} . \end{aligned} \tag{17}$$

In contrast to the arithmetic mean distance and the harmonic mean distance, Eq. (17) means that the generalized mean distance not only can give more appropriate weighted classification contributions from different neighbors, but also can significantly enhance the positive influence on classification from the neighbors with very small distances by adjusting the values of p. Thus, as a good distance measure between a pair of sample groups, the generalized mean distance is introduced to design our proposed GMDKNN.

3.2. Motivation of GMDKNN

In the KNN-based classification, the main problem is the sensitiveness to the neighborhood size k in the phase of making classification decision, especially in the situations of the small training sample size with the existing outliers.

In most of the KNN-based methods, the single and uniform value of k for all testing samples from all the classes generates the issue that the classification performance is very sensitive to k (Pan et al., 2017). An unsuitable value of k can seriously degrade the classification accuracy. Although the LMKNN can well overcome the sensitiveness to k in some degree, its classification performance is still not good when the values of k are very large or too small due to the classification decision that is designed by one local mean vector of k nearest neighbors in each class. Since the multi-local mean vectors of samples can represent the different geometrical distributions of data, LMPNN employs k local mean vectors based on k nearest neighbors to obtain the sum of weighted distances between k local mean vectors and each query sample for classification (Gou et al., 2014). The classification performance of LMPNN is robust to k, but the identical weight mechanism for each local mean vector in each class can lose some useful discrimination information of samples. Using k local mean vectors per class, the harmonic mean distance between local mean vectors and each query sample is calculated and used for classification in the MLMKHNN, in order to further overcome the sensitivity of k (Pan et al., 2017). The good classification of MLMKHNN is achieved by using the superior characteristic of the harmonic mean distance of multi-local mean vectors, but the negative influence from the existing outliers is not well degraded and the positive influence from true neighbors is not further enhanced in the region of k-neighborhood. This shortcoming of the harmonic mean distance can be well overcome by extending the harmonic mean distance to the generalized mean distance in the KNN-based classification (Luukka, 2009; 2010; Oh & Kwak, 2016). Thus, we can design multi-generalized mean distances of k local mean vectors of k nearest neighbors from each class in the classifier design.

Using the superiority of multi-local mean vectors and the characteristic of the generalized mean distance, we develop a novel generalized mean distance-based k-nearest neighbor classifier (GMDKNN). In the proposed GMDKNN, just like in LMPNN and MLMKHNN, k local mean vectors per class based on k nearest neighbors of each query sample are first calculated. These multi-local mean vectors can fully represent the local sample distributions of each class and consider the different distributions in the case of the uniform k for all classes. Since the generalized mean distance has the characteristic of differently handling the data samples, the generalized mean distance between the given query sample and its k local mean vectors can well represent the different weight contributions of these local mean vectors for classification. Furthermore, the different generalized mean distances computed by multi-local mean vectors can further effectively reflect the classification contributions from the local mean vectors in k-neighborhood. So multi-generalized mean distances based on k local mean vectors in each class are computed in the proposed GMDKNN. The multi-generalized mean distances can further consider the local sample distributions for different classes and degrade the sensitiveness to k. Using the multi-generalized mean distances, we novelly design a new generalized mean distance called a nested generalized mean distance in each class as the classification decision rule of the proposed method.

As a result, the main good characteristics of the proposed GMDKNN are derived from multi-local mean vectors, multi-generalized mean distances and the nested generalized mean distance for each class. It has been proven that multi-local mean vectors can overcome the sensitivity of k (Gou et al., 2014; Pan et al., 2017). Furthermore, multi-generalized mean distances can fully provide different weight contributions from each local mean vector, and the nested generalized mean distance as the classification decision rule can further reduce the sensitivity of k and well enhance the classification importance of each local mean vector. The superiorities of multi-generalized mean distances and the nested generalized mean distance are further analyzed in details in next Section 3.4.

3.3. Description of GMDKNN

In the proposed GMDKNN method, the class label of a given query sample y is achieved as follows:

(a) Search k nearest neighbors of y from the set X^j in each class ω_j, denoted as $X_j^{NN} = \{x_{ij}^{NN} \in \mathbb{R}^d\}_{i=1}^k$. Note that the k nearest neighbors are sorted in an ascending order in terms of their Euclidean distances to y.

(b) Calculate k local mean vectors using the top $i(1 \le i \le k)$ nearest neighbors of y from each class ω_j according to Eq. (4), denoted as $U_j^{NN} = \{u_{ij}^{NN} \in \mathbb{R}^d\}_{i=1}^k$. The corresponding Euclidean distances to y are indicated as the set $D_j^{NN} = \{d(y, u_{1j}^{NN}), d(y, u_{2j}^{NN}), \cdots, d(y, u_{kj}^{NN})\}$.

(c) Calculate k generalized mean distances using the first r $(1 \le r \le k)$ distances in the set D_j^{NN} for the class ω_j. The generalized mean distance of the first r distances from the first r local mean vectors to y for each class is computed as

$$g(y, U_{rj}^{NN}) = \left(\frac{1}{r}\sum_{i=1}^{r}\left(d(y, u_{ij}^{NN})\right)^p\right)^{\frac{1}{p}}, \quad j = 1, 2, \ldots, m, \ 1 \le r \le k, \tag{18}$$

where $U_{rj}^{NN} = \{u_{ij}^{NN} \in \mathbb{R}^d\}_{i=1}^r$ and k generalized mean distances in class ω_j are denoted as $g^j = \{g(y, U_{1j}^{NN}), g(y, U_{2j}^{NN}) \cdots g(y, U_{kj}^{NN})\}$.

(d) Design a new nested generalized mean distance based on k generalized mean distances in Eq. (18) for each class as

$$G(y, g^j) = \left(\frac{1}{k}\sum_{r=1}^{k}\left(g(y, U_{rj}^{NN})\right)^p\right)^{\frac{1}{p}}, \ j = 1, 2, \ldots, m. \tag{19}$$

(e) Classify the query sample y to the class that has the minimum nested generalized mean distance as

$$\omega = \arg\min_{\omega_j} G(y, g^j), \ j = 1, 2, \ldots, m \ . \tag{20}$$

It is worthy to note that the proposed GMDKNN degrades into MLMKHNN when $k = 1$ and $p = -1$, because the nested generalized mean distance in Eq. (19) is same as the generalized mean distance in Eq. (18) when the nested generalized mean distance is only computed by the local mean vector u_{1j}^{NN} that is equal to x_{1j}^{NN}. Unlike MLMKHNN in Section 2.3, the proposed GMDKNN extends the harmonic mean distance to the generalized mean distance, designs multi-generalized mean distances and the nested generalized mean distance for each class, and chooses the class-specific nested generalized mean distances as the classification decision rule. At the same time, when $k = 1$ GMDKNN is the same as LMPNN, LMKNN, WKNN and KNN. The pseudo codes of the proposed GMDKNN method are intuitively summarized in Algorithm 1.

3.4. Analysis of GMDKNN

From a methodological perspective, we further analyze the rationale and advantages of the proposed GMDKNN in this subsection. The proposed method has three significant characteristics: multi-local mean vectors, multi-generalized mean distances and a nested generalized mean distance from each class for a query sample.

First of all, k local mean vectors based on k nearest neighbors of y from each class are obtained. As pointed out in LMPNN (Gou et al., 2014) and MLMKHNN (Pan et al., 2017), multi-local mean vectors instead of only one local mean vector of k nearest neighbors can well represent the different local sample distributions of different classes and reduce the negative effect of the existing outliers, so as to overcome the sensitivity of k and obtain satisfactory classification performance.

Secondly, it has proven that the generalized mean distance can enhance the classification importance from samples with small distances and overcome the negative from the outliers in the region of k-neighborhood (Oh & Kwak, 2016). Just like the harmonic mean distance (Pan et al., 2017), the generalized mean distance of multi-local mean vectors per class can strengthen the more contribution of a closer subclass that corresponds to a closer local mean vector. The derivative of the generalized mean distance $g(y, U_{rj}^{NN})$ with

Algorithm 1 The generalized mean distance-based k-nearest neighbor algorithm.

```
Require:
  $X = \{x_i \in \mathbb{R}^d\}_{i=1}^{n}$: the training set with $n$ training samples.
  $C = \{\omega_1, \omega_2, \ldots, \omega_m\}$: theclass set with $m$ classes.
  $X^j = \{x_{ij} \in \mathbb{R}^d\}_{i=1}^{n_j}$: the training subset from the class $\omega_j$ with $n_j$ training samples.
  $y$: the given query sample.
  $k$: the neighborhood size.
Ensure:
  Predict the class label of the query sample $y$ bythe minimum nested generalized mean distance among all the classes.
  Step 1: Calculate the distances of training samples from each class $\omega_j$ to $y$.
  for $i = 1$ to $n_j$ do
      $d(y, x_{ij}) = \sqrt{(y - x_{ij})^T (x - x_{ij})}$
  end for
  Step 2: Find $k$ nearest neighbors of $y$from class$\omega_j$ by the top $k$ closest distances$d(y, x_{ij})$, say $X_j^{NN} = \{x_{ij}^{NN} \in \mathbb{R}^d\}_{i=1}^{k}$.
  Step 3: Obtain $k$ local mean vectors and the corresponding distances to $y$ in class $\omega_j$,denoted as $U_j^{NN} = \{u_{ij}^{NN} \in \mathbb{R}^d\}_{i=1}^{k}$ and $D_j^{NN} = \{d(y, u_{1j}^{NN}), d(y, u_{2j}^{NN}), \cdots, d(y, u_{kj}^{NN})\}$, respectively.
  for $i = 1$ to $k$ do
      $u_{ij}^{NN} = \frac{1}{i}\sum_{q=1}^{i} x_{qj}^{NN}$,
      $d(y, u_{ij}^{NN}) = \sqrt{(y - u_{ij}^{NN})^T (y - u_{ij}^{NN})}$
  end for
  Step 4: Calculate $k$ generalized mean distances based on $k$ local mean vectors in the set $D_j^{NN}$ for class $\omega_j$, denoted as $g^j = \{g(y, U_{1j}^{NN}), g(y, U_{2j}^{NN}) \cdots g(y, U_{kj}^{NN})\}$.
  for $r = 1$ to $k$ do
      $g\left(y, U_{rj}^{NN}\right) = \left(\frac{1}{r}\sum_{i=1}^{r}\left(d(y, u_{ij}^{NN})\right)^p\right)^{\frac{1}{p}}$
  end for
  Step 5: Calculate the nested generalized mean distance based on $k$ harmonic mean distances in the set $g^j$ for the class $\omega_j$.
      $G(y, g^j) = \left(\frac{1}{k}\sum_{r=1}^{k}\left(g(y, U_{rj}^{NN})\right)^p\right)^{\frac{1}{p}}$
  Step 6: Classify the query sample $y$ to the class that has the minimum nested generalized mean distance among $m$ classes.
      $\omega = \arg\min_{\omega_j} G(y, g^j),\ j = 1, 2, \ldots, m.$
```

respect to $d(y, u_{ij}^{NN})$ using Eqs. (17) and (18) can prove this property. The value of $\frac{\partial g(y, U_{rj}^{NN})}{\partial d(y, u_{ij}^{NN})}$ represents the weight of the classification contribution from the local mean vector u_{ij}^{NN} with the distance $d(y, u_{ij}^{NN})$ as follows:

$$\begin{aligned}\frac{\partial g(y, U_{rj}^{NN})}{\partial d(y, u_{ij}^{NN})} &= \frac{\partial\left(\left(\frac{1}{r}\sum_{i=1}^{r}\left(d(y, u_{ij}^{NN})\right)^p\right)^{\frac{1}{p}}\right)}{\partial d(y, u_{ij}^{NN})}\\ &= \frac{1}{r}\times\left(\frac{1}{r}\sum_{i=1}^{r}\left(d(y, u_{ij}^{NN})\right)^p\right)^{\frac{1-p}{p}}\left(d(y, u_{ij}^{NN})\right)^{(p-1)}\\ &= \frac{1}{r}\times\left(g(y, U_{rj}^{NN})\right)^{(1-p)}\left(d(y, u_{ij}^{NN})\right)^{(p-1)}\\ &= \frac{1}{r}\times\left(\frac{g(y, U_{rj}^{NN})}{d(y, u_{ij}^{NN})}\right)^{(1-p)},\\ &j = 1, 2, \ldots, m,\ 1 \le r \le k\ , 1 \le i \le r.\end{aligned} \tag{21}$$

According to Eq. (21), it is clear that a closer local mean vector can make the weighted contribution for classification $\frac{\partial g(y, U_{rj}^{NN})}{\partial d(y, u_{ij}^{NN})}$ larger. When making classification decision, the generalized mean distance $g(y, U_{rj}^{NN})$ gives more classification contribution of the local mean vector that has a smaller distance to the query sample y, and also the closer local mean vector can significantly make the generalized mean distance to be smaller in terms of Eq. (18) when $p < 1$. Meantime, the weighted contribution from each local mean vector can be differently adjusted by the values of p in the generalized mean. Moreover, for multi-generalized mean distances based on the first r local mean vectors in terms of Eq. (21), we can clearly observe that one local mean vector per class can have different weight contributions in multi-generalized mean distances. For example, the second local mean vector u_{2j}^{NN} in generalized mean distances $g(y, U_{2j}^{NN})$ and $g(y, U_{3j}^{NN})$ has the weights $\frac{1}{2}\times\left(\frac{g(y, U_{2j}^{NN})}{d(y, u_{2j}^{NN})}\right)^{(1-p)}$ and $\frac{1}{3}\times\left(\frac{g(y, U_{3j}^{NN})}{d(y, u_{2j}^{NN})}\right)^{(1-p)}$ as the classification contribution, respectively. This means that multi-generalized mean distances can fully utilize the multi-local mean vectors and further reduce the sensitiveness of the neighborhood size k.

Thirdly, we design a new generalized mean distance that is called the nested generalized mean distance for each class using multi-generalized mean distances in the proposed GMDKNN method. Just like the generalized mean distance, we consider the weighted contribution in the nested generalized mean distance from each local mean vector u_{ij}^{NN} with the distance $d(y, u_{ij}^{NN})$. The derivative value of $G(y,\ g^j)$ with respect to $d(y, u_{ij}^{NN})$ (i.e. $\frac{\partial G(y, g^j)}{\partial d(y, u_{ij}^{NN})}$) only represents the weight of the classification contribution from the local mean vector u_{ij}^{NN} in a generalized mean distance $g(y, U_{rj}^{NN})$. Using Eqs. (18) and (19), $\frac{\partial G(y, g^j)}{\partial d(y, u_{ij}^{NN})}$ is reformulated as

$$\frac{\partial G(y, g^j)}{\partial d(y, u_{ij}^{NN})} = \frac{\partial G(y, g^j)}{\partial g(y, U_{rj}^{NN})}\times\frac{\partial g(y, U_{rj}^{NN})}{\partial d(y, u_{ij}^{NN})},\quad 1 \le r \le k \text{ and } 1 \le i \le r. \tag{22}$$

Just like the derivations in Eq. (21), $\frac{\partial G(y, h^j)}{\partial g(y, U_{rj}^{NN})}$ in Eq. (22) can be easily calculated as follows:

$$\begin{aligned}\frac{\partial G(y, g^j)}{\partial g(y, U_{rj}^{NN})} &= \frac{\partial\left(\left(\frac{1}{k}\sum_{r=1}^{k}\left(g(y, U_{rj}^{NN})\right)^p\right)^{\frac{1}{p}}\right)}{\partial g(y, U_{rj}^{NN})}\\ &= \frac{1}{k}\times\left(\frac{1}{k}\sum_{r=1}^{k}\left(g(y, U_{rj}^{NN})\right)^p\right)^{\frac{1-p}{p}}\left(g(y, U_{rj}^{NN})\right)^{(p-1)}\\ &= \frac{1}{k}\times\left(G(y, g^j)\right)^{(1-p)}\left(g(y, U_{rj}^{NN})\right)^{(p-1)}\\ &= \frac{1}{k}\times\left(\frac{G(y, g^j)}{g(y, U_{rj}^{NN})}\right)^{(1-p)},\ 1 \le r \le k.\end{aligned} \tag{23}$$

According to Eqs. (21) and (23), Eq. (22) can be rewritten as

$$\begin{aligned}\frac{\partial G(y, g^j)}{\partial d(y, u_{ij}^{NN})} &= \frac{1}{kr}\times\left(\frac{G(y, g^j)}{g(y, U_{rj}^{NN})}\right)^{(1-p)}\times\left(\frac{g(y, U_{rj}^{NN})}{d(y, u_{ij}^{NN})}\right)^{(1-p)}\\ &= \frac{1}{kr}\times\left(\frac{G(y, g^j)}{d(y, u_{ij}^{NN})}\right)^{(1-p)},\\ &1 \le r \le k \text{ and } 1 \le i \le r.\end{aligned} \tag{24}$$

According to Eq. (24), the new nested generalized mean distance in Eq. (19) has the similar properties with the generalized mean distance in Eq. (11). Moreover, since the nested generalized mean distance is obtained by multi-generalized mean distances and a multi-generalized mean distance is also computed by a different number of local mean vectors, a local mean vector u_{ij}^{NN} can make different weighted contributions for favorable classification according to $g(y, U_{rj}^{NN})$ in the nested generalized mean distance when $i \le r$. And also, the weighted classification contribution from local mean vector u_{ij}^{NN} can be differently adjusted and strengthened by the values of p. Thus, the property in the nested generalized mean distance can further overcome the sensitivity of k with good classification performance, even if a single uniform value of k is for each class. These superiorities in the proposed GMDKNN are experimentally verified by extensive experiments in Section 4.

3.5. Differences between GMDKNN and LMKNN, LMPNN, MLMKHNN

In this subsection, we further clearly emphasize the differences between the proposed GMDKNN and the related methods including LMKNN, LMPNN and MLMKHNN.

1. GMDKNN vs. LMKNN: In the LMKNN, the local mean vector of k nearest neighbors from each class is used for making classification decision so as to well overcome the outliers. But the sensitiveness to the neighborhood size k still degrades the classification performance. However, in the proposed GMDKNN the multi-local mean vectors of k nearest neighbors and multi-generalized mean distances of multi-local mean vectors and a nested generalized mean distance are employed for the KNN-based classification with robustness and effectiveness.
2. GMDKNN vs. LMPNN: both LMPNN and GMDKNN use multi-local mean vectors of k nearest neighbors from each class. In the LMPNN, pseudo nearest neighbor based on multi-local mean vectors from each class is employed as the classification decision to well overcome the sensitiveness to k with the satisfactory performance. However, the classification performance of LMPNN is still affected by the identical weight mechanism for each local mean vector. In the proposed GMDKNN, multi-generalized mean distances of multi-local mean vectors are used for computing a nested generalized mean distance in each class, and class-specific nested generalized mean distance as the classification decision is employed for good classification.
3. GMDKNN vs. MLMKHNN: MLMKHNN and GMDKNN adopt multi-local mean vectors of k nearest neighbors from each class. In the MLMKHNN, the only one harmonic mean distance of multi-local mean vectors from each class is calculated for classifying the given query sample with robustness to k. Unlike the MLMKHNN, the GMDKNN introduces generalized mean distance, which is the general form of the harmonic mean distance. Moreover, GMDKNN calculates multi-generalized mean distances of multi-local mean vectors and then uses them to design a nested generalized mean distance in each class as the classification decision rule. Compared to MLMKHNN, the proposed GMDKNN not only overcomes the sensitiveness to k, but also differently handles the classification importance of nearest samples to fully use their pattern discrimination.

4. Experiments

In this section, we conduct the extensive experiments on many data sets to verify the classification performance of the proposed GMDKNN method. GMDKNN is compared with KNN, WKNN, PNN, LMKNN, LMPNN and MLMKHNN in light of the classification recognition rate that is one of the most effective measures in pattern recognition (Li et al., 2008; Mitani & Hamamoto, 2006).

Table 1
The real data sets in the experiments from both UCI and KEEL repositories.

Data sets	Datebases	Features	Samples	Classes	Testing samples
Sonar	UCI	60	208	2	66
Letter	UCI	16	20,000	26	6000
Climate	UCI	18	540	2	218
Vowel	UCI	10	528	11	176
Image	UCI	19	2310	7	770
Iono	UCI	34	351	2	90
Park	UCI	22	195	2	65
Seed	UCI	7	210	3	60
Glass	UCI	9	146	2	53
Musk	UCI	166	476	2	162
Thyroid	UCI	21	7200	3	2485
Robot	UCI	4	5456	4	1826
Opt	UCI	64	5620	10	2620
Landsat	UCI	36	6435	6	2159
Pen	UCI	16	10,992	10	3662
Vehicle	UCI	18	846	4	213
Wdbc	UCI	30	569	2	150
Wine	UCI	13	178	3	58
Libras	UCI	90	360	15	120
Balance	UCI	4	625	3	181
Newth	KEEL	5	215	3	70
Tae	KEEL	5	151	3	60
Phoneme	KEEL	5	5404	2	1504
Spambase	KEEL	57	4597	2	1500
Band	KEEL	19	365	2	120
Dermat	KEEL	34	358	6	115
Ring	KEEL	20	7400	2	3400
Texture	KEEL	40	5500	11	1650

4.1. Experiment 1

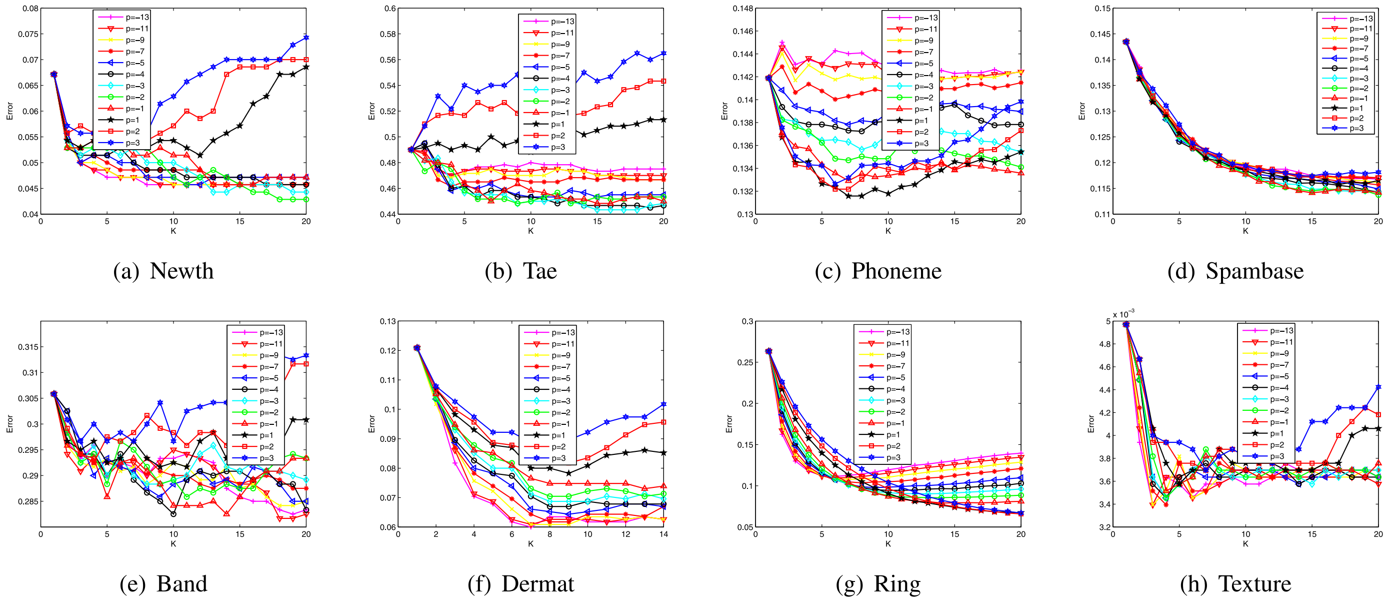
In this subsection, the comparative experiments of the competing methods on many real UCI and KEEL data sets are first carried out, in order to demonstrate the effectiveness and robustness of the proposed GMDKNN method.

4.1.1. Real data sets

The twenty-eight real data sets in the experiments are utilized, and we first summarize their main information shown in Table 1. Among them, twenty and eight real date sets are from UCI (Lichman, 2013) and KEEL (Alcala-Fdez et al., 2011) repositories, respectively. Note that ‘Climate Model Simulation Crashes’, ‘Image segmentation’, ‘Ionosphere’, ‘Parkinsons’, ‘Robot Navigation’, ‘Opt digits’, ‘Landsat Satellite’, ‘Pen digits’, ‘Libras Movement’, ‘Newthyroid’, and ‘Dermatology’ are abbreviated as ‘Climate’, ‘Image’, ‘Iono’, ‘Park’, ‘Robot’, ‘Opt’, ‘Landsat’, ‘Pen’, ‘Libras’, ‘Newth’ and ‘Dermat’, respectively. These data sets have different characteristics in the numbers of samples, features and classes. They have a wide range of the total numbers of samples that are varied from 146 to 20,000 and correspond to different multi-class classification problems, so as to comprehensively demonstrate the classification performance of the proposed method. Besides, the numbers of samples on most of the data sets are very small, these data sets can be well used to verify the classification performance in the small sample size cases. These experiments are done by a ten hold-out validation procedure. In other words, each real data set is randomly divided into training and testing samples ten times and we get the different training and testing sets on each data set each time. The numbers of testing samples on each data set are also listed in Table 1. The average classification rates with 95% confidence on ten divisions of each data set are obtained as the final classification results for each method. obtained as the final classification results for each method.

4.1.2. Experimental results on real data sets

In the experiments, since the proposed GMDKNN method has two parameters that are p in the generalized mean dis-



(a) Newth (b) Tae (c) Phoneme (d) Spambase

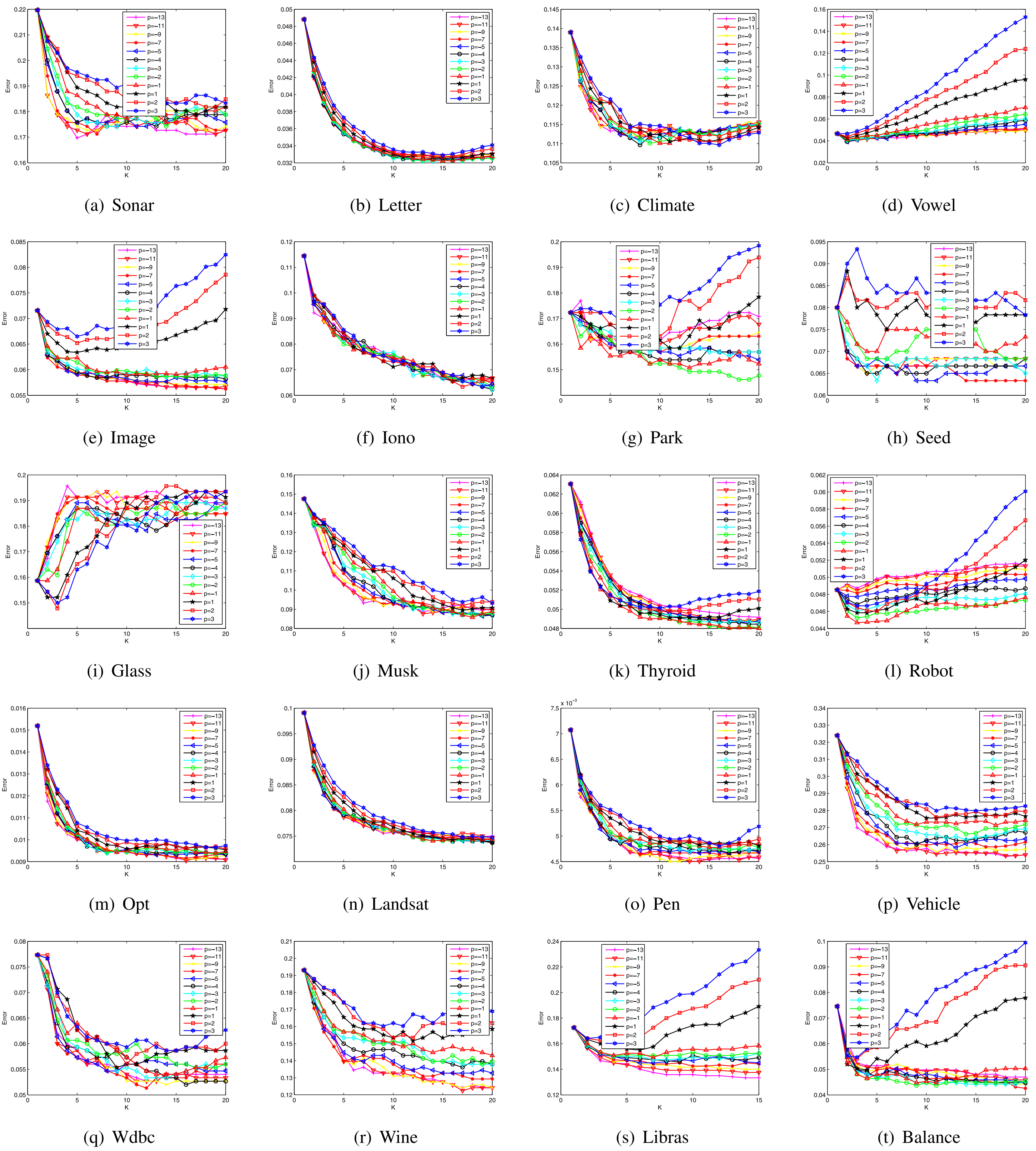
(e) Band (f) Dermat (g) Ring (h) Texture

Fig. 1. The classification results of the proposed method via the different values of p and k on each KEEL data set in terms of the classification error rates.

tance and the neighborhood size k, we first evaluate its classification performance via the different values of p and k on each real data set in terms of the classification error rates. The values of p are varied in the preseted range $\{-13, -11, -9, -7, -5, -4, -3, -2, -1, 1, 2, 3\}$. The values of k are preseted by the range from 1 to 20 with a step 1 on all real data sets except Libras and Dermat. The values of k are varied from 1 to 15 on Libras and from 1 to 14 on Dermat with a step 1. The classification results of the proposed method with varying the values of p and k are shown in Figs. 1 and 2. We can clearly see that the classification error rates of the proposed method are not nearly influenced by different values of p on Letter, Climate, Iono, Opt, Landsat and Spambase, but are significantly influenced by different values of p on the other real data sets except Glass. And most importantly, we can observe that the better classification results of the proposed method on most of the real data sets are obtained at the negative values of p. This experimental fact verifies the theoretical analysis from Eq. (12) and holds the characteristic of differently handling the data samples in the generalized mean distances. Furthermore, we can observe that the proposed method with different values of p always achieve the satisfactory classification results at large values of k, and also its classification error rates on most of the data sets slowly increase or keep to be stable when k becomes large. This means the proposed GMDKNN can choose more nearest neighbors to get the promising classification and has robustness to k.

To evaluate the classification performance of the competing methods with regard to the sensitivity of the neighborhood size k, the comparative experiments on real data sets are carried out with varying the values of k by means of the classification error rates. The classification results of the competing methods via the neighborhood size k on UCI and KEEL real data sets are shown in Figs. 3 and 4. Note that the value of p in the parenthesis in the caption of each subfigure of Figs. 3 and 4 is obtained from Figs. 1 and 2 when the proposed GMDKNN can almost obtain the better classification results among the range of p. From the classification results in Figs. 3 and 4, we can obviously observe that the proposed GMDKNN method almost performs better than the other methods with different values of k, particular in the case of the large values of k. On most of the real data sets, the classification error rates of GMDKNN first fastly descend at small values of k and then nearly tend to be stable when values of k are large. Such experimental phenomenon is also obtained in LMPNN and MLMKHNN. This implies that the proposed GMDKNN, LMPNN and MLMKHNN have more robustness to the different values of k with good classification. The reason can be that three methods employ multi-local mean vectors of more nearest neighbors for designing the classification decision rules to obtain this advantage. However, we can see that the classification error rates of KNN, WKNN, PNN and LMKNN almost increase with an increase of the neighborhood size k, and their classification performance is very sensitive to k. Thus, the classification results in Figs. 3 and 4 fully demonstrate the good classification performance of the proposed method with less sensitiveness to the neighborhood size k, and the proposed method is superior to the other comparative methods in most cases.

After the evaluations of the competing methods via the neighborhood size k, the maximal classification accuracy rates of each method with the corresponding 95% confidence intervals and the optimized values of k on KEEL and UCI real data sets are displayed in Tables 2 and 3, respectively. Note that the optimized values of k for all the methods are listed in the parentheses, and the best classification performance among all the methods on each data set is denoted in bold type. As shown in two tables, it is clear that the proposed GMDKNN method almost outperforms the other state-of-art methods on all the real data sets. It can be seen that the classification results of LMPNN, MLMKHNN and GMDKNN are very better than KNN, WKNN, LMKNN and PNN. The superior performance of LMPNN, MLMKHNN and GMDKNN is obtained by mainly employing the multi-local mean vectors of k nearest neighbors per class. And most importantly, we can observe from the classification accuracy rates in Tables 2 and 3 that GMDKNN is better than MLMKHNN on each real data sets, because the proposed GMDKNN introduces the multi-generalized mean distances based on multi-local mean vectors and the nested generalized mean distance based on multi-generalized mean distances in each class. Meanwhile, it can be noted that MLMKHNN performs better than LMPNN on 22 out of 28 real data sets and GDMKNN almost performs better than LMPNN on each real data set, because the generalized mean distance can fully use local mean vectors and focus on the more reliable local mean vectors with very smaller distances to every testing sample in each class. The fact from the superior performance of the proposed GMDKNN over MLMKHNN and LMPNN



(a) Sonar (b) Letter (c) Climate (d) Vowel

(e) Image (f) Iono (g) Park (h) Seed

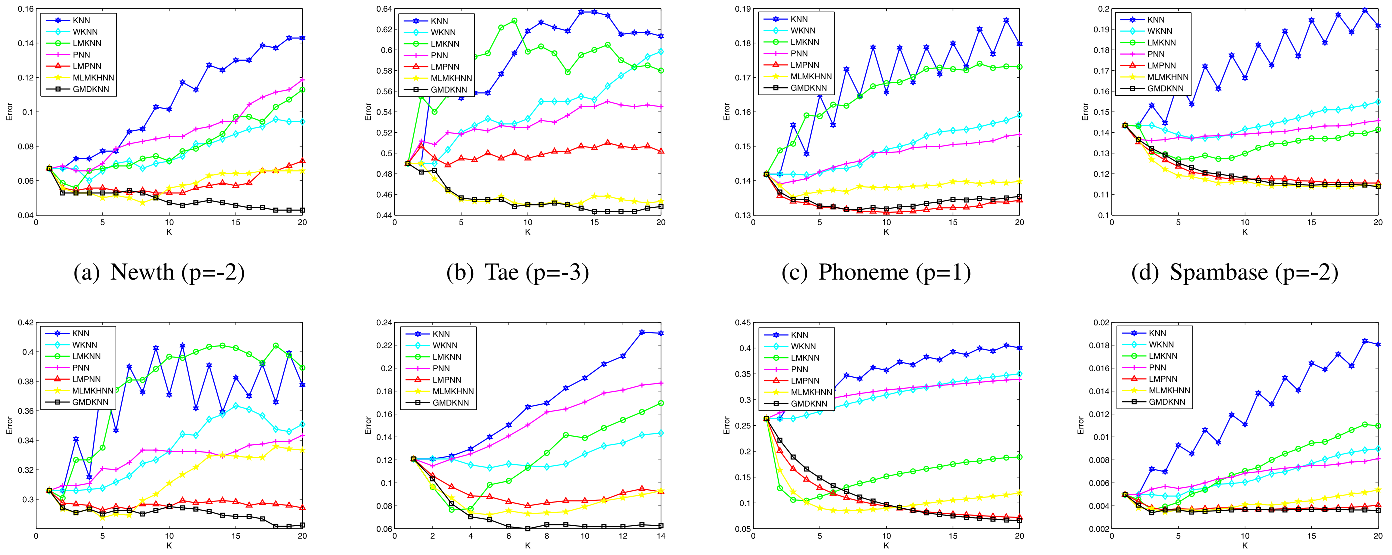
(i) Glass (j) Musk (k) Thyroid (l) Robot

(m) Opt (n) Landsat (o) Pen (p) Vehicle

(q) Wdbc (r) Wine (s) Libras (t) Balance

Fig. 2. The classification results of the proposed method via the different values of p and k in the generalized mean on each UCI data set in terms of the classification error rates.

and the superior performance of MLMKHNN over LMPNN implies that the generalized mean distance can not only make more classification contribution from the local mean vectors with very small distances in each class, but also make one local mean vector play different importance in different multi-generalized mean distances for classification, in order that the nested generalized mean distance per class can get more classification information from multi-local mean vectors in each class. From the point of view of the neighborhood size k, the values of k corresponding to the best classification in the proposed GMDKNN method are nearly larger than the ones in the other competing methods on all the real data sets. This indicates that GMDKNN can utilize more nearest neighbors to get different multi-generalized mean distances for favorable classification.



(a) Newth (p=-2) (b) Tae (p=-3) (c) Phoneme (p=1) (d) Spambase (p=-2)

(e) Band (p=-2) (f) Dermat (p=-13) (g) Ring (p=2) (h) Texture (p=-1)

Fig. 3. The classification results of each method via k on each KEEL data set in terms of the classification error rates.

Table 2
The highest classification accuracy rates (%) of each method with the corresponding standard deviations (stds) and values of k in the parentheses on all the KEEL data sets (the best recognition performance among competing methods on each data set is described in bold-face).

Data set	KNN	WKNN	LMKNN	PNN	LMPNN	MLMKHNN	GMDKNN
Newth	93.29 ± 1.79 (1)	94.00 ± 2.21 (4)	94.43 ± 2.97 (3)	93.43 ± 2.79 (3)	94.71 ± 2.86 (7)	95.29 ± 2.86 (8)	**95.71 ± 2.61 (18)**
Tae	51.00 ± 6.95 (1)	51.00 ± 6.95 (1)	51.00 ± 6.95 (1)	51.00 ± 6.95 (2)	51.17 ± 5.72 (4)	55.00 ± 5.77 (17)	**55.67 ± 6.20 (15)**
Phoneme	85.81 ± 0.65 (1)	85.84 ± 0.59 (4)	85.81 ± 0.65 (1)	86.10 ± 0.76 (2)	**86.92 ± 0.60 (10)**	86.48 ± 0.87 (3)	86.84 ± 0.79 (7)
Spambase	85.65 ± 0.82 (1)	86.27 ± 0.90 (7)	87.30 ± 0.28 (5)	86.39 ± 0.85 (3)	88.45 ± 0.50 (19)	**88.63 ± 0.62 (17)**	**88.63 ± 0.62 (20)**
Band	69.42 ± 2.42 (1)	69.42 ± 2.42 (1)	69.42 ± 2.17 (2)	69.42 ± 2.42 (1)	70.75 ± 2.62 (5)	71.25 ± 3.20 (5)	**71.83 ± 3.55 (18)**
Dermat	87.91 ± 2.70 (1)	88.70 ± 1.79 (5)	92.35 ± 2.20 (3)	88.52 ± 1.78 (2)	92.00 ± 2.35 (7)	92.78 ± 1.84 (5)	**94.00 ± 1.61 (7)**
Ring	73.67 ± 0.45 (1)	73.67 ± 0.45 (1)	89.55 ± 0.52 (4)	73.67 ± 0.45 (1)	92.80 ± 0.23 (20)	91.54 ± 0.59 (7)	**93.41 ± 0.26 (20)**
Texture	99.50 ± 0.18 (1)	99.52 ± 0.19 (4)	99.64 ± 0.17 (3)	99.50 ± 0.18 (1)	99.64 ± 0.15 (4)	99.65 ± 0.11 (4)	**99.66 ± 0.12 (3)**

Through all the experimental results on real data sets above, the effectiveness and robustness of the proposed GMDKNN method are well demonstrated. Compared to the state-of-art methods, the best classification performance of GMDKNN is almost achieved with less sensitiveness of the neighborhood size k among the comparative methods.

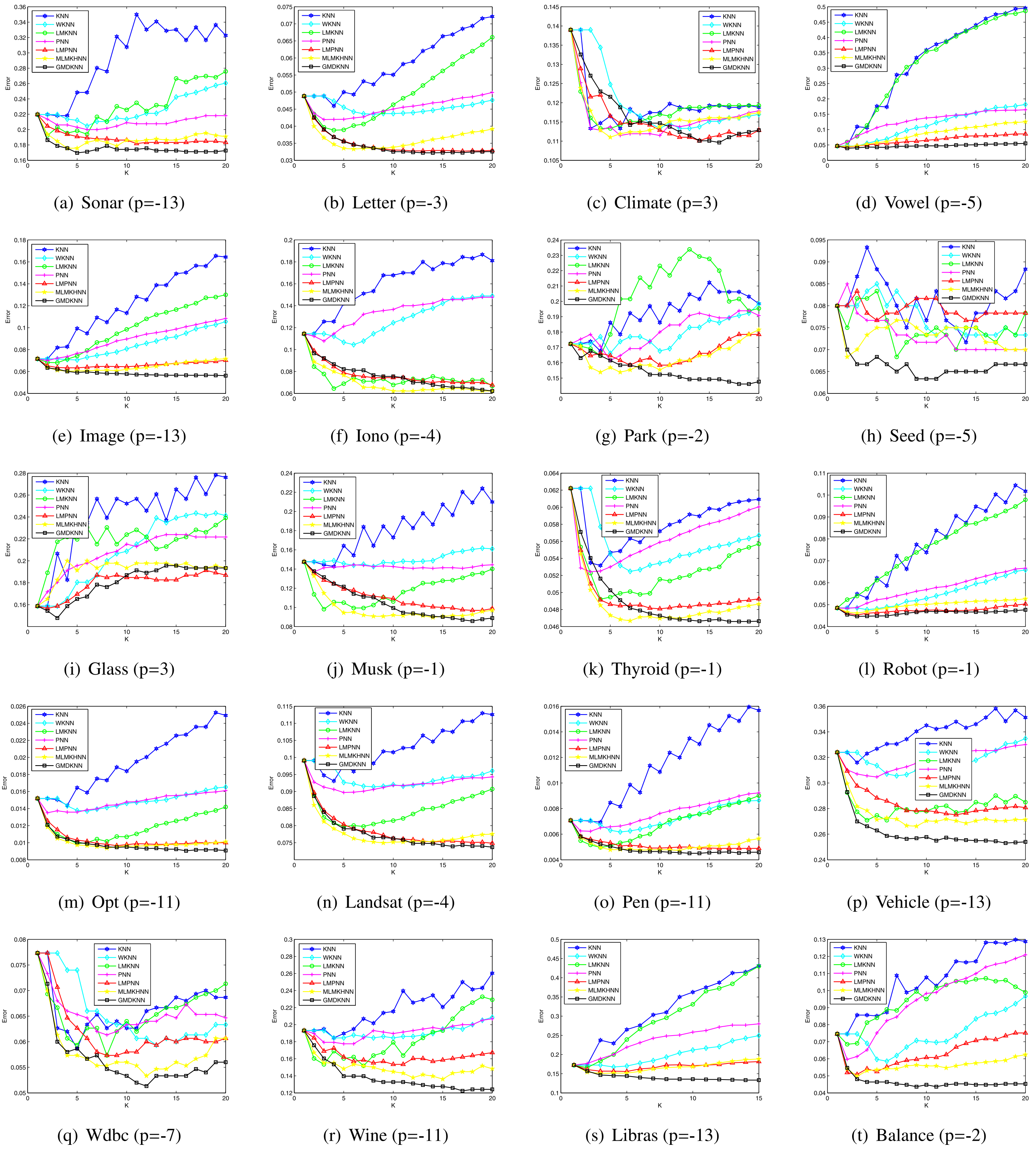
4.2. Experiment 2

To further verify the proposed GMDKNN method, we carry out the experiments on four synthetic data sets, in comparisons with KNN, WKNN, PNN, LMKNN, LMPNN and MLMKHNN in this subsection.

4.2.1. Synthetic data sets

We first briefly describe the four used artificial data sets in the experiments. These data sets are called Rotated checkerboard, Cone-torus, I-$\bar{\wedge}$ and I-6I, respectively. It should be noted that Rotated checkerboard and Cone-torus data sets are generated by the matlab codes taken from the web site http://pages.bangor.ac.uk/~mas00a/activities/artificial_data.htm. Rotated checkerboard is a two-dimensional data model with two classes. In the Rotated checkerboard data, a checkerboard is tilted to b with side of the black squares equal to a (s.t. $0 < a < 1$), where $a = 0.2$ and $b = -\pi/3$ in the experiments. Fig. 5(a) shows an example of the Rotated checkerboard data with 1000 samples. Cone-torus is also an artificial two-dimensional data model that has three classes (i.e., cone, half-torus, Gaussian). Three classes have a cone, half a torus, and a normal distributions with different prior probabilities, each of which in the experiments is set as 0.3, 0.4 and 0.3, respectively. As an example, Fig. 5(b) shows a Cone-torus data set with 1000 samples. I-$\bar{\wedge}$ and I-6I are eight-dimensional Gaussian data models with two classes, each of which has the same prior probability, and are randomly generated by employing the singular normal density functions. Two data models are generated as follows:

1. I-$\bar{\wedge}$ data set:
   $u_1 = 0, \quad \Sigma_1 = I_8,$
   $u_2 = [2.86, 3.10, 0.84, 0.84, 1.64, 1.08, 0.26, 0.01]^T,$
   $\Sigma_2 = \text{diag}[10.41, 11.06, 0.12, 0.22, 1.49, 1.77, 0.35, 2.73].$
2. I-6I data set:
   $u_1 = 0, \quad u_2 = 0$
   $\Sigma_1 = I_8, \quad \Sigma_2 = 6I_8.$



(a) Sonar (p=-13)
(b) Letter (p=-3)
(c) Climate (p=3)
(d) Vowel (p=-5)
(e) Image (p=-13)
(f) Iono (p=-4)
(g) Park (p=-2)
(h) Seed (p=-5)
(i) Glass (p=3)
(j) Musk (p=-1)
(k) Thyroid (p=-1)
(l) Robot (p=-1)
(m) Opt (p=-11)
(n) Landsat (p=-4)
(o) Pen (p=-11)
(p) Vehicle (p=-13)
(q) Wdbc (p=-7)
(r) Wine (p=-11)
(s) Libras (p=-13)
(t) Balance (p=-2)

Fig. 4. The classification results of each method via k on each UCI data set in terms of the classification error rates.

In the experiments, the numbers of the randomly generated training samples and testing samples are 600 and 400 on Rotated checkerboard, 230 and 170 on Cone-torus, 100 and 100 on both I-$\bar{\lambda}$ and I-6I, respectively. For each artificial data model, the training and testing sets are randomly generated each time. We use a ten-trials hand-out procedure to get the average classification results with 95% confidence as the final classification performance evaluation on the four artificial data models.

4.2.2. Experimental results on synthetic data sets

First of all, we investigate the classification performance of the proposed GMDKNN method with varying the values of p and k in terms of the classification error rate on the synthetic data sets. Note that the values of p in the generalized mean distance are

Table 3
The highest classification accuracy rates (%) of each method with the corresponding standard deviations (stds) and values of k in the parentheses on all the UCI data sets (the best recognition performance among competing methods on each data set is described in bold-face).

Data set	KNN	WKNN	LMKNN	PNN	LMPNN	MLMKHNN	GMDKNN
Sonar	78.18 ± 2.49 (3)	79.55 ± 3.21 (6)	80.76 ± 3.57 (2)	80.00 ± 3.10 (6)	81.82 ± 3.64 (11)	82.42 ± 2.78 (4)	**83.03 ± 3.90 (5)**
Letter	95.41 ± 0.37 (4)	95.65 ± 0.32 (8)	96.12 ± 0.21 (4)	95.81 ± 0.31 (3)	96.73 ± 0.27 (17)	96.67 ± 0.25 (6)	**96.78 ± 0.25 (14)**
Climate	88.67 ± 1.08 (3)	88.67 ± 1.02 (10)	88.67 ± 0.49 (4)	88.85 ± 0.99 (6)	88.99 ± 0.53 (14)	88.90 ± 0.91 (5)	**89.04 ± 0.50 (16)**
Vowel	95.34 ± 2.41 (1)	95.34 ± 2.41 (1)	95.34 ± 2.41 (1)	95.34 ± 2.41 (1)	95.68 ± 2.47 (2)	95.51 ± 2.46 (2)	**96.08 ± 2.79 (2)**
Image	92.84 ± 0.63 (1)	92.95 ± 0.58 (5)	93.19 ± 0.86 (3)	93.00 ± 0.59 (2)	93.66 ± 0.77 (4)	93.97 ± 0.72 (4)	**94.38 ± 0.70 (20)**
Iono	88.56 ± 2.72 (1)	89.56 ± 2.04 (6)	93.56 ± 2.08 (4)	89.22 ± 2.10 (3)	93.22 ± 1.92 (20)	**93.78 ± 2.23 (19)**	**93.78 ± 1.90 (20)**
Park	83.23 ± 3.28 (4)	83.38 ± 4.02 (5)	83.54 ± 5.13 (4)	83.85 ± 3.10 (5)	84.15 ± 4.17 (7)	84.62 ± 3.16 (4)	**85.38 ± 3.64 (18)**
Seed	92.83 ± 3.05 (14)	92.67 ± 2.11 (11)	93.17 ± 3.64 (7)	93.00 ± 2.70 (13)	92.33 ± 2.25 (5)	93.17 ± 1.46 (2)	**93.67 ± 2.19 (9)**
Glass	84.13 ± 6.49 (1)	84.13 ± 6.49 (1)	84.13 ± 6.49 (1)	84.13 ± 6.49 (1)	84.13 ± 4.56 (2)	84.13 ± 6.49 (2)	**85.22 ± 5.31 (2)**
Musk	85.74 ± 2.54 (4)	85.74 ± 3.07 (8)	90.19 ± 1.68 (3)	86.30 ± 1.51 (2)	90.31 ± 1.91 (18)	91.05 ± 1.49 (14)	**91.42 ± 2.03 (18)**
Thyroid	94.68 ± 0.23 (4)	94.75 ± 0.26 (7)	95.08 ± 0.30 (4)	94.76 ± 0.24 (3)	95.19 ± 0.22 (10)	95.33 ± 0.24 (7)	**95.34 ± 0.29 (17)**
Robot	95.15 ± 0.43 (1)	95.22 ± 0.53 (4)	95.15 ± 0.43 (1)	95.16 ± 0.45 (2)	95.47 ± 0.32 (3)	95.39 ± 0.41 (3)	**95.53 ± 0.40 (3)**
Opt	98.56 ± 0.20 (4)	98.63 ± 0.20 (6)	99.01 ± 0.16 (5)	98.65 ± 0.15 (2)	99.03 ± 0.17 (9)	99.06 ± 0.16 (8)	**99.10 ± 0.15 (16)**
Landsat	90.69 ± 0.40 (4)	90.86 ± 0.53 (8)	92.02 ± 0.43 (5)	91.03 ± 0.54 (5)	92.52 ± 0.49 (20)	92.51 ± 0.29 (9)	**92.63 ± 0.37 (20)**
Pen	99.31 ± 0.14 (4)	99.38 ± 0.16 (6)	99.50 ± 0.09 (4)	99.38 ± 0.15 (3)	99.51 ± 0.11 (16)	99.53 ± 0.11 (12)	**99.55 ± 0.15 (13)**
Vehicle	68.40 ± 2.53 (3)	69.44 ± 2.22 (7)	72.91 ± 1.98 (4)	69.53 ± 2.59 (5)	72.49 ± 2.26 (13)	73.38 ± 2.63 (9)	**74.69 ± 3.01 (18)**
Wdbc	94.13 ± 1.29 (5)	94.07 ± 1.19 (13)	94.27 ± 1.30 (8)	93.87 ± 1.53 (8)	94.27 ± 1.55 (8)	94.67 ± 1.72 (12)	**94.87 ± 1.94 (12)**
Wine	81.55 ± 5.27 (4)	81.55 ± 6.30 (4)	84.83 ± 5.06 (7)	82.24 ± 6.08 (5)	84.66 ± 6.21 (10)	86.38 ± 3.85 (15)	**87.76 ± 6.52 (17)**
Libras	82.75 ± 1.36 (1)	83.25 ± 1.64 (4)	83.42 ± 2.79 (2)	82.75 ± 1.36 (1)	84.42 ± 2.49 (5)	84.92 ± 2.24 (3)	**86.67 ± 2.81 (14)**
Balance	92.54 ± 1.23 (1)	94.14 ± 1.83 (6)	93.15 ± 2.02 (2)	94.03 ± 1.76 (2)	94.92 ± 1.58 (3)	95.03 ± 1.33 (3)	**95.64 ± 1.44 (9)**

set as the range $\{-13, -11, -9, -7, -5, -4, -3, -2, -1, 1, 2, 3\}$, and the values of k are preseted from 1 to 20 with a step 1. The classification error rates of the proposed method via different values of p and k are displayed in Fig. 6. As shown in the figure, we can see that the classification error rates of the proposed method are influenced by different values of p, and the better classification results are always obtained at the large values of k on Cone-torus, I-$\bar{\wedge}$ and I-6I. These experimental fact means that the generalized mean distance can differently handle the data samples and its proper values of p is data dependent in the proposed method.

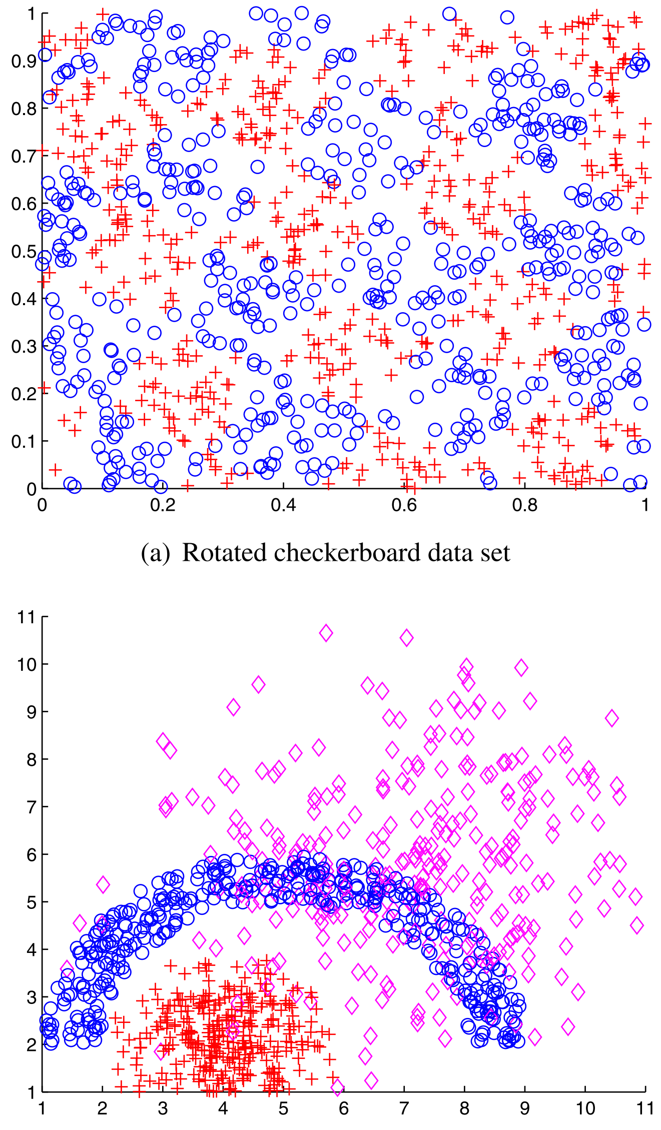
The comparative classification performance of KNN, WKNN, PNN, LMKNN, LMPNN, MLMKHNN and GMDKNN are studied on Rotated checkerboard, Cone-torus, I-$\bar{\wedge}$ and I-6I in terms of classification error rates with varying the neighborhood size k. Note that the value of p in the parenthesis in the caption of each subfigure is optimally obtained from Fig. 6 with the better classification results. The classification results of each method via the different values of k on each artificial data set is shown in Fig. 7. From the experimental results in Fig. 7, the proposed GMDKNN is almost best among the competing methods with different values of k and is least sensitive to k. We can observe that LMPNN, MLMKHNN and GMDKNN obtain the similar classification pattern: their classification error rates first decrease at small values of k, and then nearly keep stable. This experimental fact from three methods is due to the multi-local mean vectors used for classification. Nevertheless, the classification error rates of KNN, WKNN, PNN and LMKNN quickly ascend with increasing the values of the neighborhood size k. Therefore, according to the experimental results on four artificial data sets, the less sensitivity of the neighborhood size k with satisfactory classification performance in the proposed GMDKNN is demonstrated.

4.3. Experiment 3

As we know, it is more difficult to perform good classification under the situation of the real data sets with noises. In this subsection, the effectiveness and robustness of the proposed GMDKNN method are further comparatively evaluated on four KEEL real data sets with attribute noises in terms of the classification error rates.

4.3.1. Noise data sets

The four noise data sets from the KEEL Repository (Alcala-Fdez et al., 2011) in Table 4 are 'Wine', 'Sonar', 'Image segmentation' and 'Pen digits', which are processed with attribute noise by the schema in Wu (2004). In the experiments, they are abbreviated as 'Wine_n', 'Sonar_n', 'Image_n' and 'Pen_n', respectively. In the four noise data sets, they are introduced by 5% noisy in the samples. That is to say, the 5% of the samples in each data set are chosen approximately and the values of one attribute for these examples are assigned by the random values between the minimum and maximum of the domain of the attribute following a uniform distribution. It should be noted that the four noise data sets have



(a) Rotated checkerboard data set

(b) Cone-torus data set

Fig. 5. The examples of two artificial data sets.

Table 4
The noise data sets from KEEL repository.

Data sets	Datebases	Features	Samples	Classes	Testing samples
Wine_n	KEEL	13	178	3	35
Sonar_n	KEEL	60	208	2	41
Image_n	KEEL	19	2310	7	462
Pen_n	KEEL	16	10992	2	2199

been downloaded in the KEEL Repository. On each noise data set, training and testing samples have been divided five times in the KEEL Repository and the numbers of testing samples are shown in Table 4. The average classification results of five divisions on each noise data set are achieved as the finial classification evaluation of each competing method with 95% confidence.

4.3.2. Experimental results on noise data sets

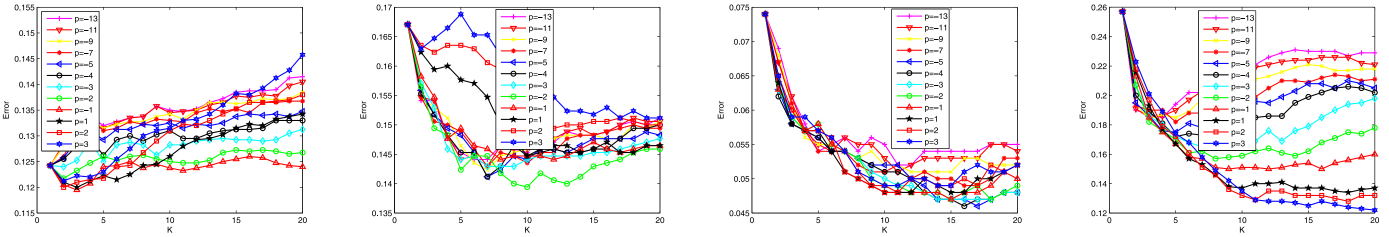
The classification performance of the proposed GMDKNN is first explored by varying different values of p and k on four noise data sets. Note that the preseted ranges of the values of p and k are the same as the Sections 4.1 and 4.2. The classification error rates of the proposed method via p and k are shown in Fig. 8. We can see from Fig. 8 that the classification results of the proposed method are influenced by the values of p. This means that the generalized mean distance differently handles the data samples in terms of adjusting the values of p.

After determining the proper values of p in the proposed method, the comparative experiments of the competing methods are further conducted by varying the numbers of the neighborhood size k. The classification error rates of the competing methods via k are listed in Fig. 9. Note that the optimal values of p in the parentheses in the captions of subfigures in Fig. 9 are obtained from Fig. 8. From the classification results in Fig. 9, we can obviously observe that the proposed GMDKNN method almost achieve the better classification performance than the other methods with different values of k. We can also see that GMDKNN, LMPNN and MLMKHNN perform better and are less sensitive to k than KNN, WKNN, PNN and LMKNN, and they obtain the satisfactory classification results when the values of k are large. Thus, it can be concluded that the proposed GMDKNN can well overcome the sensitivity of the neighborhood size k with good classification performance by using more nearest neighbors.

The maximal classification accuracy rates with the corresponding optimized values of k and 95% confidence intervals for each competing method on the four noise data sets are listed in Table 5. It is clear that the proposed GMDKNN method almost outperforms the other methods. We can also see that GMDKNN, LMPNN and MLMKHNN are always better than KNN, WKNN, PNN and LMKNN on the noise data sets. The possible reason is that GMDKNN, LMPNN and MLMKHNN fully use the multi-local mean vectors based on k nearest neighbors in each class. Furthermore, compared to the other methods, we can observe that the proposed GMDKNN always employs more nearest neighbors to obtain better classification.

4.4. Experiment 4

In this subsection, we further verify the classification performance of the proposed GMDKNN method on the time series data sets with the relative high dimensionalities in the time series mining tasks (Tran et al., 2017).



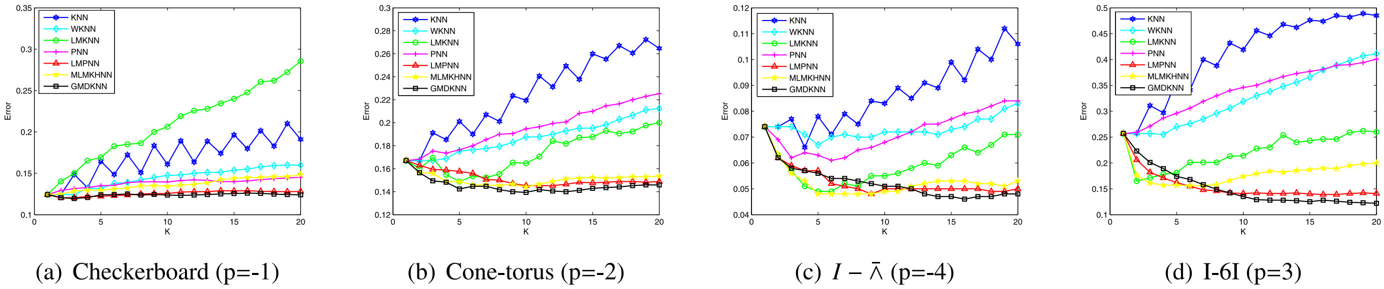
(a) Checkerboard

(b) Cone-torus

(c) $I - \bar{\Lambda}$

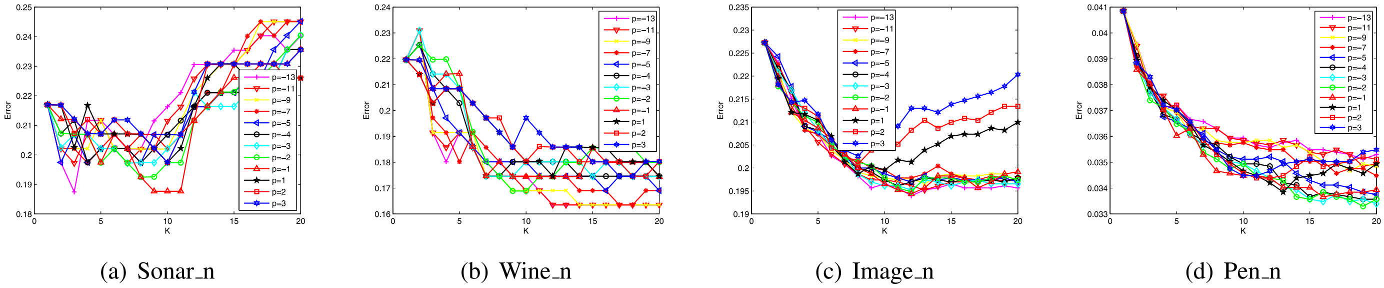
(d) I-6I

Fig. 6. The classification results of the proposed method via the different values of p and k on synthetic data sets.



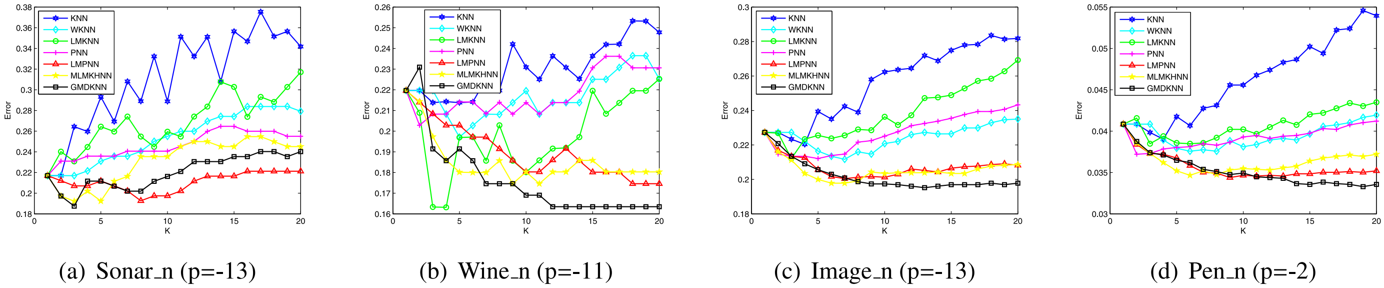
(a) Checkerboard (p=-1) (b) Cone-torus (p=-2) (c) $I - \bar{\lambda}$ (p=-4) (d) I-6I (p=3)

Fig. 7. The classification results of the competing methods via the neighborhood size k on synthetic data sets.



(a) Sonar_n (b) Wine_n (c) Image_n (d) Pen_n

Fig. 8. The classification results of the proposed method via the different values of p and k on noise data sets.



(a) Sonar_n (p=-13) (b) Wine_n (p=-11) (c) Image_n (p=-13) (d) Pen_n (p=-2)

Fig. 9. The classification results of the competing methods via the neighborhood size k on noise data sets.

Table 5
The highest classification accuracy rates (%) of each method with the corresponding standard deviations (stds) and values of k in the parentheses on all the KEEL data sets with noises (the best recognition performance among competing methods on each data set is described in bold-face).

Data set	KNN	WKNN	LMKNN	PNN	LMPNN	MLMKHNN	GMDKNN
Sonar_n	78.30 ± 7.63 (1)	78.30 ± 7.63 (1)	78.30 ± 7.63 (1)	78.30 ± 7.63 (1)	80.73 ± 4.68 (8)	80.77 ± 6.32 (3)	**81.25 ± 7.40 (3)**
Wine_n	78.62 ± 7.91 (3)	80.29 ± 7.32 (5)	**83.68 ± 6.12 (4)**	79.71 ± 8.66 (2)	82.54 ± 6.25 (18)	82.54 ± 5.56 (9)	83.65 ± 5.61 (12)
Image_n	77.97 ± 0.83 (4)	78.83 ± 1.24 (7)	78.79 ± 1.20 (3)	78.79 ± 2.47 (5)	79.96 ± 1.88 (7)	80.22 ± 2.11 (6)	**80.61 ± 1.30 (12)**
Pen_n	96.10 ± 1.09 (4)	96.24 ± 1.14 (8)	96.15 ± 1.17 (6)	96.28 ± 1.00 (2)	96.56 ± 1.11 (9)	96.53 ± 1.17 (6)	**96.63 ± 1.08 (19)**

4.4.1. UCR time series data sets

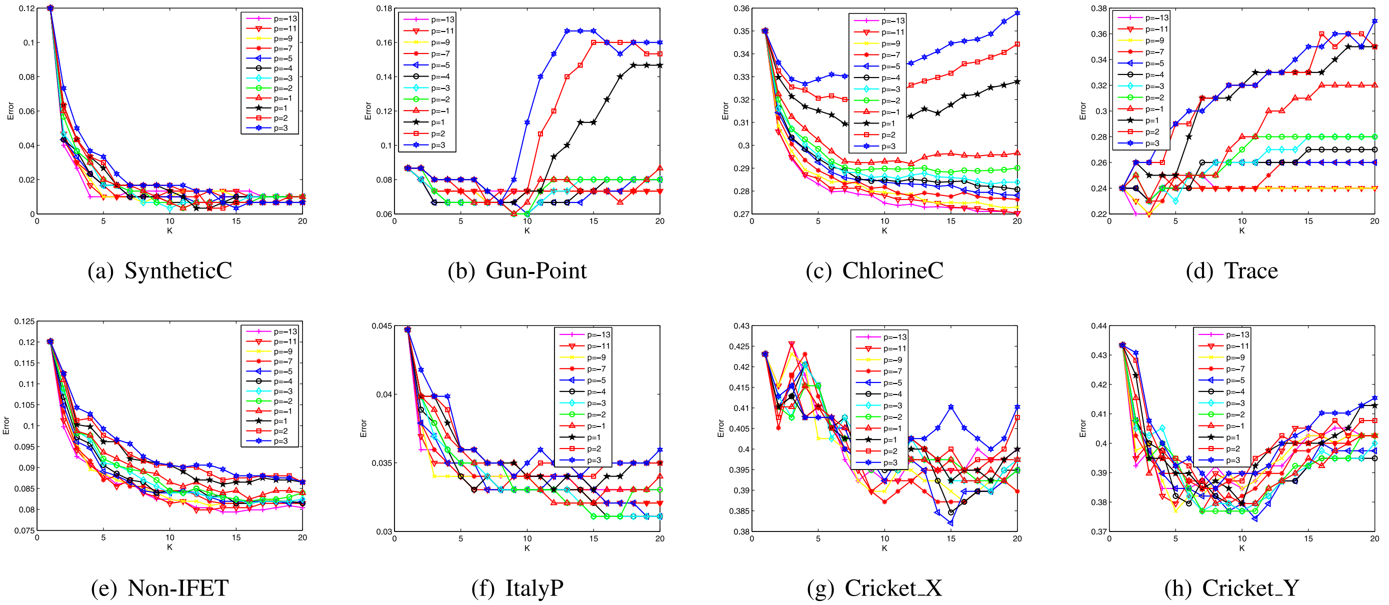
In the experiments, eight time series data sets taken from UCR time series classification archive (Chen et al., 2015) are used and their main properties are shown in Table 6. It should be noted that the sets of training and testing samples on each data set were fixed and directly downloaded from the website (Chen et al., 2015). And for simple notations in the experiments, the 'Synthetic Control', 'ChlorineConcentration', 'Non-Invasive Fetal ECG Thorax2' and 'ItalyPowerDemand' time series data sets are abbreviated as 'SyntheticC', 'ChlorineC', 'Non-IFET' and 'ItalyP', respectively. All experiments of the competing methods on these time series data sets are conducted by varying the neighborhood size k in terms of the classification recognition rates.

4.4.2. Experimental results on time series data sets

The classification performance of the proposed GMDKNN on eight time series data sets are studied by varying the values of p and k. The preseted ranges of the values of p and k on time series data sets are the same as on real, synthesis and noise data sets above. The classification error rates of the proposed method with different values of p and k are illustrated in Fig. 10. We can see from the classification results in Fig. 10 the classification performance of GMDKNN is influenced by the values of p and its satisfactory classification performance are always achieved at the negative values of p. This experimental fact implies that the proper values of p can well differently handle data samples and verifies the theoretical analysis above. Furthermore, the better classification results

Table 6
The time series data sets used in the experiments.

Data set	Training samples	Testing samples	Time series lengths	Classes
SyntheticC	300	300	60	6
Gun-Point	50	150	150	2
ChlorineC	467	3840	166	3
Trace	100	100	275	4
Non-IFET	1800	1965	750	42
ItalyP	67	1029	24	2
Cricket_X	390	390	300	12
Cricket_Y	390	390	300	12



(a) SyntheticC (b) Gun-Point (c) ChlorineC (d) Trace

(e) Non-IFET (f) ItalyP (g) Cricket_X (h) Cricket_Y

Fig. 10. The classification results of the proposed method via the different values of p and k on time series data sets.

of the proposed method are always obtained at the large values of the neighborhood size k.

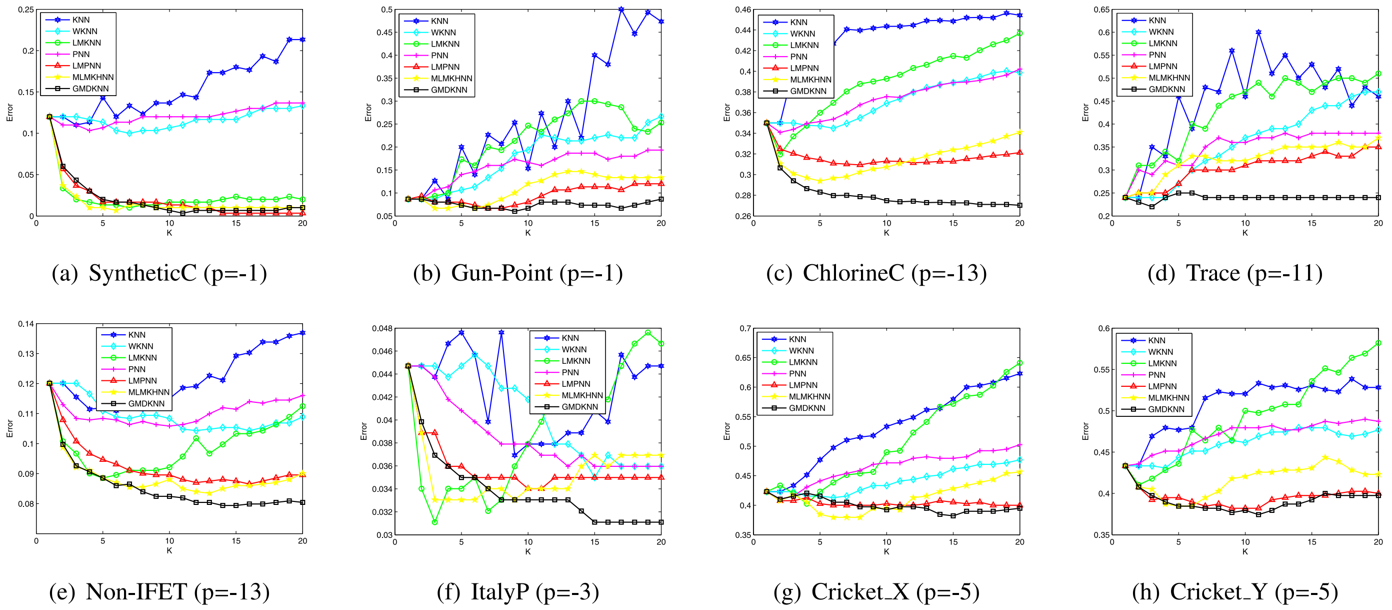
The experimental classification error rates of the competing methods with varying the neighborhood size k are shown in Fig. 11. As displayed in Fig. 11, unlike the compared KNN, WKNN, PNN, LMKNN, LMPNN and MLMKHNN, the classification error rates of the proposed GMDKNN nearly decreases at first at small values of k, and then nearly tends to be stable when the values of k are large. It can be seen that GMDKNN almost performs better than the other competing methods, especially in the case of the larger value of k. In addition, we can observe that the performance of GMDKNN, LMPNN and MLMKHNN that use the multi-local mean vectors for classification is always better than KNN, WKNN, PNN and LMKNN. Thus, we can conclude from these experimental results in Fig. 11 that the proposed method are more robust to neighborhood size k with better classification recognition rates than the competing methods. These experimental results which the proposed GMDKNN often significantly outperforms the other methods mean that GMDKNN can employ more nearest neighbors for satisfactory classification.

In the range of the neighborhood size k, the maximal classification accuracy rate of each competing method on each time series data set is illustrated in Table 7. Note that the best classification accuracy among the competing methods on each data set is described in bold-face. It is obvious that the proposed GMDKNN almost has the highest recognition performance among the competing methods in the case of the large values of the neighborhood size k. Therefore, the experimental results demonstrated that the proposed method is effective in these time series data sets with high dimensionalities.

In a word, the classification performance of the proposed GMDKNN is extensively verified under the situations of the real data sets, the artificial data sets, the noise data sets and the time series data sets. Through experimental results above, several observations can be summarized as follows:

(a) The proposed GMDKNN method can well overcome the sensitivity of the neighborhood size k with satisfactory classification performance.
(b) The proposed GMDKNN method can use more nearest neighbors to get both multi-local mean vectors and multi-generalized mean distances to calculate the nested generalized mean distance in each class for making favorable classification decision.
(c) The methods (i.e., GMDKNN, LMPNN and MLMKHNN) using multi-local mean vectors in each class always perform better than other KNN-based methods.
(d) The generalized mean can differently handle data samples as the good similarity measure for the KNN-based classification (Luukka, 2009; 2010; Oh & Kwak, 2016). .
(e) The effectiveness and robustness of the proposed GMDKNN method has been held in many classification problems and it is a promising classifier in pattern recognition.

Through several observation from the experimental analyses above, it can be indicated that the multi-local mean vectors, multi-generalized mean distances and the nested generalized mean distances based on nearest samples could well improve the effectiveness and robustness of the KNN-based classification, especially in the small sample size cases with the existing outliers.



(a) SyntheticC (p=-1)

(b) Gun-Point (p=-1)

(c) ChlorineC (p=-13)

(d) Trace (p=-11)

(e) Non-IFET (p=-13)

(f) ItalyP (p=-3)

(g) Cricket_X (p=-5)

(h) Cricket_Y (p=-5)

Fig. 11. The classification results of the competing methods via the neighborhood size *k* on time series data sets.

Table 7
The highest classification accuracy rates (%) of each method with the values of *k* in the parentheses on all the time series data set. (the best recognition performance among the competing methods on each data set is described in bold-face).

Data set	KNN	WKNN	LMKNN	PNN	LMPNN	MLMKHNN	HMDKNN
SyntheticC	89.00 (3)	90.00 (7)	99.00 (7)	**89.67** **(4)**	99.67 (14)	99.33 (7)	**99.67** **(11)**
Gun-Point	91.33 (1)	91.33 (1)	91.33 (1)	91.33 (1)	93.33 (7)	93.33 (3)	**94.00** **(9)**
ChlorineC	65.00 (1)	65.52 (6)	68.05 (2)	65.91 (2)	69.06 (8)	70.63 (5)	**72.97** **(20)**
Trace	76.00 (1)	76.00 (1)	76.00 (1)	76.00 (1)	76.00 (1)	76.00 (1)	**78.00** **(3)**
Non-IFET	88.91 (6)	89.57 (12)	91.15 (5)	89.41 (10)	91.35 (16)	92.65 (13)	**92.06** **(14)**
ItalyP	96.31 (9)	96.50 (15)	**96.89** **(3)**	96.40 (13)	96.60 (10)	96.70 (3)	**96.89** **(15)**
Cricket_X	57.69 (1)	58.72 (6)	59.74 (4)	58.97 (2)	60.00 (6)	**62.05** **(6)**	61.79 (15)
Cricket_Y	56.67 (1)	56.92 (4)	58.97 (2)	56.67 (1)	61.79 (9)	61.54 (5)	**62.56** **(11)**

5. Conclusions

In this article, the proposed GMDKNN is motivated by using multi-local mean vectors, multi-generalized mean distances and the nested generalized mean distances on the basis of *k* nearest neighbors in each class. The main purpose of GMDKNN is to overcome the sensitivity of the neighborhood size *k* and improve the KNN-based classification performance. In the proposed method, the nested generalized mean distance calculated by the multi-generalized mean distances in each class is designed for the KNN-based classification decision. The GMDKNN method has the merits of both LMPNN and MLMKHNN, because GMDKNN, LMPNN and MLMKHNN use multi-local mean vectors and MLMKHNN employs the harmonic mean distance that is a special case of the generalized mean distance used in GMDKNN. These merits capture the more discrimination information from *k* nearest neighbors for the good classification. Furthermore, multi-generalized mean distances and the corresponding nested generalized mean distance in each class not only fully utilize the multi-local mean vectors, but also allow one local mean vector to make the different weighted contributions for classification. And the weighted contributions from local mean vectors can be differently handled according to data samples and adaptively adjusted by the values of *p* in the generalized mean. The proposed method can employ more nearest neighbors for the favorable classification and has less sensitiveness to the values of *k*. We compare GMDKNN to the state-of-art KNN-based methods on many data sets. The extensively experimental results show that the proposed method obtains an improvement in classification performance. Thus, the proposed GMDKNN is a promising KNN-based classifier in pattern recognition.

In future, we will plan to apply multi-generalized mean distances and the nested generalized mean distance to some related KNN-based classification methods to overcome the sensitivity of the neighborhood size *k*. Both generalized mean distances and multi-local mean vectors can also be further utilized to address the sensitiveness to the existing outliers in pattern classification. Moreover, in terms of the detailed analyses, the proposed GMDKNN could be well applied in many real-world classification tasks in the field of expert and intelligence systems. For instance, we can use generalized mean distance as a rejection measure to solve the problems of the rejection recognition, which means that some query samples are not probably from any candidate classes and they should be rejected in pattern classification. Besides, multi-generalized mean distances and/or multi-local mean vectors can be integrated with the other methods for good classification.

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (Grant Nos. 61502208, 61762021 and 61402122), Natural Science Foundation of Jiangsu Province of China (Grant Nos. BK20150522 and BK20170558) and Research Foundation for Talented Scholars of Jiangsu University (Grant No. 14JDG037), China Postdoctoral Science Foundation (Grant No. 2015M570411), Open Foundation of Artificial Intelligence Key Laboratory of Sichuan Province (Grant No. 2017RYJ04), Natural Science Foundation of Guizhou Province (No.[2017]1130), and Research Foundation of North Minzu University (Grant No. 2018XYZDX10).

References

Alcala-Fdez, J., Fernandez, A., Luengo, J., Derrac, J., Garcia, S., Sanchez, L., & Herrera, F. (2011). KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing, 17*(2–3), 255–287. http://www.keel.es/

Bicego, M., & Loog, M. (2016). Weighted k-nearest neighbor revisited. In *23rd International conference on pattern recognition (ICPR)* (pp. 1642–1647). IEEE.

Bo, C., Lu, H., & Wang, D. (2017). Weighted generalized nearest neighbor for hyperspectral image classification. *IEEE Access, 5*, 1496–1509.

Bulut, F., & Amasyali, M. F. (2017). Locally adaptive k parameter selection for nearest neighbor classifier: One nearest cluster. *Pattern Analysis and Applications, 20*, 415–425.

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., & Batista, G. (2015). The UCR time series classification archive. www.cs.ucr.edu/~eamonn/time_series_data/.

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*, 21–27.

Domeniconi, C., & Gunopulos, D. (2002). Locally adaptive metric nearest-neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(9), 1281–1285.

Dudani, S. A. (1976). The distance-weighted k-nearest neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics, 6*(4), 325–327.

Fang, X., Lu, Y., Li, Z., Yu, L., & Chen, Y. (2014). Kernel representation-based nearest neighbor classifier. *Optik-International Journal for Light and Electron Optics, 125*(10), 2320–2326.

Garcia-Pedrajas, N., Castillo, J. A. R. D., & Cerruela-Garcia, G. (2017). A proposal for local k values for k-nearest neighbor rule. *IEEE Transactions on Neural Networks and Learning Systems, 28*, 470–475.

Gou, J., Du, L., Zhang, Y., & Xiong, T. (2012). A new distance-weighted k-nearest neighbor classifier. *Journal of Information & Computational Science, 9*(6), 1429–1436.

Gou, J., Xiong, T., & Kuang, Y. (2011). A novel weighted voting for k-nearest neighbor rule. *Journal of Computers, 6*(5), 833–840.

Gou, J., Yi, Z., Du, L., & Xiong, T. (2012). A local mean-based k-nearest centroid neighbor classifier. *The Computer Journal, 55*(9), 1058–1071.

Gou, J., Zhan, Y., Rao, Y., Shen, X., Wang, X., & He, W. (2014). Improved pseudo nearest neighbor classification. *Knowledge-Based Systems, 70*, 361–375.

Jung, W. H., & Lee, S. G. (2017). An arrhythmia classification method in utilizing the weighted KNN and the fitness rule. *IRBM, 38*, 138–148.

Li, B., Chen, Y. W., & Chen, Y. Q. (2008). The nearest neighbor algorithm of local probability centers. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 38*(1), 141–153.

Li, P., Gou, J., & Yang, H. (2017). The distance-weighted k-nearest centroid neighbor classification. *Journal of Intelligent Information Hiding and Multimedia Signal Processing, 8*, 611–622.

Li, W., Du, Q., Zhang, F., & Hu, W. (2015). Collaborative representation-based nearest neighbor classifier for hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters, 12*, 389–393.

Lichman, M. (2013). *UCI machine learning repository*. Irvine, CA: University of California, School of Information and Computer Science. http://archive.ics.uci.edu/ml.

Luukka, P. (2009). Classification based on fuzzy robust PCA algorithms and similarity classifier. *Expert Systems with Applications, 36*(4), 7463–7468.

Luukka, P. (2010). Nonlinear fuzzy robust PCA algorithms and similarity classifier in bankruptcy analysis. *Expert Systems with Applications, 37*(12), 8296–8302.

Ma, H., Gou, J., Ou, W., Zeng, S., Rao, Y., & Yang, H. (2017). A new nearest neighbor classifier based on multi-harmonic mean distances. *IEEE 2017 international conference on security, pattern analysis, and cybernetics. Dec. 15–17, 2017, Shenzhen, Guangdong, China*.

Ma, H., Gou, J., Wang, X., Ke, J., & Zeng, S. (2017). Sparse coefficient-based *k*-nearest neighbor classification. *IEEE Access, 5*, 16618–16634.

Mitani, Y., & Hamamoto, Y. (2006). A local mean-based nonparametric classifier. *Pattern Recognition Letters, 27*, 1151–1159.

Oh, J., & Kwak, N. (2016). Generalized mean for robust principal component analysis. *Pattern Recognition, 54*, 116–127.

Pan, Z., Wang, Y., & Ku, W. (2017). A new k-harmonic nearest neighbor classifier based on the multi-local means. *Expert Systems with Applications, 67*, 115–125.

Tran, T. M., Le, X. M. T., Vinh, V. T., Nguyen, H. T., & Nguyen, T. M. (2017). A weighted local mean-based k-nearest neighbors classifier for time series. In *Proceedings of the 9th International Conference on Machine Learning and Computing* (pp. 157–161). ACM.

Vidyarthi, A., & Mittal, N. (2016). AVNM: A voting based novel mathematical rule for image classification. *Computer Methods and Programs in Biomedicine, 137*, 195–201.

Weinberger, K. Q., & Saul, L. K. (2009). Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research, 10*, 207–244.

Wu, X., et al. (2004). Error detection and impactsensitive instance ranking in noisy datasets. In *Proceedings of 19th national conference on artificial intelligence (AAAI-2004), San Jose, CA*.

Wu, X., Kumar, V., Quinlan, J. R., & Ghosh, J. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems, 14*(1), 1–37.

Xu, Y., Zhu, Q., Fan, Z. Z., Qiu, M. N., Chen, Y., & Liu, H. (2013). Coarse to fine k nearest neighbor classifier. *Pattern Recognition Letters, 34*, 980–986.

Yang, J., Zhang, L., Yang, J. Y., & Zhang, D. (2011). From classifiers to discriminators: A nearest neighbor rule induced discriminant analysis. *Pattern Recognition, 44*, 1387–1402.

Zeng, Y., Yang, Y., & Zhao, L. (2009). Nonparametric classification based on local mean and class statistics. *Expert Systems with Applications, 36*, 8443–8448.

Zeng, Y., Yang, Y., & Zhao, L. (2009). Pseudo nearest neighbor rule for pattern classification. *Expert Systems with Applications, 36*, 3587–3595.

Zhang, J., & Yang, J. (2014). Linear reconstruction measure steered nearest neighbor classification framework. *Pattern Recognition, 47*, 1709–1720.

Zhang, N., Yang, J., & Qian, J. J. (2012). Component-based global KNN classifier for small sample size problems. *Pattern Recognition Letters, 33*(13), 1689–1694.

Zhang, S., Li, X., Zong, M., Zhu, X., & Cheng, D. (2017). Learning k for KNN classification. *ACM Transactions on Intelligent Systems and Technology, 8*(3), 1–18. 43

Zhang, S., Li, X., Zong, M., Zhu, X., & Wang, R. (2017). Efficient KNN classification with different numbers of nearest neighbors. *IEEE Transactions on Neural Networks and Learning Systems*. doi:10.1109/TNNLS.2017.2673241.

Zhong, X. F., Guo, S. Z., Gao, L., Shan, H., & Zheng, J. H. (2017). An improved KNN classification with dynamic k. In *Proceedings of the 9th international conference on machine learning and computing, ACM* (pp. 211–216).