Pattern Recognition 58 (2016) 100–109

Contents lists available at ScienceDirect

Pattern Recognition

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

Individualized learning for improving kernel Fisher discriminant analysis

Zizhu Fan [a], Yong Xu [b,*], Ming Ni [a], Xiaozhao Fang [b], David Zhang [c]

[a] School of Basic Science, East China Jiaotong University, Nanchang, China
[b] Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China
[c] Department of Computing, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

ARTICLE INFO

Article history:
Received 23 September 2012
Received in revised form
19 March 2016
Accepted 23 March 2016
Available online 6 April 2016

Keywords:
Individualized learning
KFDA
Individualized KFDA (IKFDA)
High-dimensional
Similarity measure

ABSTRACT

Kernel Fisher discriminant analysis (KFDA) is a very popular learning method for the purpose of classification. In this paper, we propose a novel learning algorithm to improve KFDA and make it very suitable for dealing with the large-scale and high-dimensional data sets. The proposed algorithm is termed individualized KFDA (IKFDA). IKFDA is based on *individualized learning*, i.e., a strategy to learn and classify the individual test samples one by one. Our approach seeks to find the appropriate training subset, referred to as *learning area*, for each individual test sample, and then employ the learning area to construct the KFDA model for the test sample. For each individual test sample, IKFDA exploits some types of similarity measures to determine a learning area that consists of the training samples that are most similar to the test sample. Compared with the traditional learning algorithms that often exploit the whole training set to construct the learning models without considering the distribution property of the test samples, IKFDA can adaptively learn the individual test samples. It is a powerful tool to deal with the real-world complicated data sets that are often very large-scale and high-dimensional, and are usually drawn from the different distributions. Extensive experiments show that the proposed algorithm can obtain good classification results.

1. Introduction

In pattern recognition and machine learning, kernel Fisher discriminant analysis (KFDA) [1,2], or kernel discriminant analysis (KDA) [3,4] has been widely used for feature extraction and classification. KFDA is one typical kind of kernel-based approaches. It is well known that in the kernel-based approaches, we adopt a nonlinear mapping to map the input data into a high-dimensional space, i.e., the feature space or kernel-induced space [5], in which the inner products can be computed by a kernel function [6,7]. We do not need to know the nonlinear mapping explicitly and the mapping is determined by our specific kernel function and its parameters in these kernel-based approaches. In essence, KFDA is a kernel-based nonlinear extension of linear discriminant analysis (LDA) [8,9]. When dealing with highly nonlinear data, KFDA often outperforms LDA in terms of the classification accuracy.

Similar to the traditional learning algorithms such as principal component analysis (PCA) [10], LDA and support vector machines (SVMs) [11,12], KFDA exploits the whole training set to construct the learning models without considering the distribution property of the test samples. Among these traditional learning algorithms including KFDA, there exists a common property. That is, these traditional learning algorithms often use only a unique common learning model to extract the features of the test samples and classify them. Here, we refer to those algorithms using a unique common learning model as *commonized learning* algorithms. The commonized learning algorithms are very suitable for the learning setting in which both the training and test sets have the same distribution. They can achieve the desirable classification effectiveness when the training and test samples are drawn from the same distribution.

However, in the real-world applications, e.g., face recognition, document classification, and webpage analysis, the data sets are usually very large-scale and high-dimensional, and may be drawn from the different distributions. Since KFDA and other commonized learning algorithms are based on the common distribution property of data sets, they often cannot well model each individual test sample. As a result, KFDA and other commonized learning algorithms often suffer from the following problems. First of all, when dealing with the large-scale and high-dimensional data sets, they usually fail to adaptively construct the learning model for

[*] Correspondence to: Bio-Computing Research Center, HIT Campus of Shenzhen University Town, Shenzhen 518055, China. Tel.: +86 755 26032458, 13640997970; fax: +86 755 26032461.
E-mail address: laterfall2@yahoo.com.cn (Y. Xu).

http://dx.doi.org/10.1016/j.patcog.2016.03.029
0031-3203/© 2016 Elsevier Ltd. All rights reserved.

each individual test sample and the learning effectiveness of them may not be desirable in general. Second, if the training set contains the noises, the learning model in the commonized learning must be severely affected by the noises and the obtained model cannot generalize well for new samples. In addition, it is usually difficult to determine the proper number of the training samples in the commonized learning. Note that the commonized learning often needs to load the whole training set to the memory at a time in the procedure of constructing the learning model. If the number of the training samples is very large, the space complexity of the commonized learning will be high when the data sets are very high-dimensional. On the other hand, when the number of the training samples is relatively small, if we exploit these samples to construct a unique common learning model, it is clear that this learning model cannot sufficiently model the total samples [13]. This usually leads to undesirable learning effectiveness.

To address the above problems from which the commonized learning algorithms including KFDA suffer, many researchers have proposed different algorithms to improve the commonized learning. There are two typical types of the improved algorithms. The first type is based on the local learning algorithms [14–16]. For each test sample, these algorithms first employ a distance measure, e.g., the Euclidean distance, to determine the nearest neighbors for the test sample. Then, they use the determined neighbors to construct one learning model and classify the test sample. The local learning algorithms can capture the local structure of the data. The second type is based on the representation approaches [17–22]. Also, these approaches learn and classify the test samples one by one. Nevertheless, unlike to the above local learning algorithms, the representation based approaches first exploit the representation based on a certain norm, e.g., L_1 norm, to represent each test sample, and then exploit the representation residual to classify the test sample. The representation based approaches such as sparse representation classification (SRC) [17] are suitable for learning the data that contain the occlusion or noise.

In fact, both the distance in the local learning algorithms and the representation residual of the representation based approaches can be viewed as the similarity measures. In the local algorithms, the larger the distance between the samples, the less similar the samples are. The representation based approaches often use the training samples belonging to each class to represent the test sample. The less representation residual generally indicates that the test sample is more similar to the samples of that class. Therefore, the representation residual can be used as a similarity measure. Notice that both the local learning algorithms and the representation based approaches exploit only one type of similarity measures when they learn or classify the test samples. When we use these algorithms to deal with the large-scale and high-dimensional data sets, employing only one type of similarity measure may not guarantee to find an appropriate training subset to learn each individual test sample. It is well-known that the training subset of a test sample is crucial to learn and classify this test sample. For a test sample, if some similarity measure does not obtain an appropriate training subset, the learning algorithm might fail. In particular, if the class label of the test sample is not included in the class label set of the obtained training subset, then, whatever the learning algorithms we use, the test sample would not be correctly classified. Therefore, in order to achieve good classification results, we should adopt the proper similarity measure scheme to determine the training subset for the test samples.

Based on the above analysis, we propose a novel learning algorithm in this paper. The proposed algorithm is based on *individualized learning*, i.e., to learn and classify the individual test samples one by one. The individualized learning algorithm can adaptively learn the individual test samples. It is very suitable for learning the large-scale and high-dimensional data sets. The proposed algorithm seeks to find the appropriate training subset, referred to as *learning area*, for each individual test sample and employ the learning area to construct the learning model for this individual test sample. In the proposed individualized learning, we learn the test samples one by one. Firstly, we employ multiple similarity measures to determine the learning area for a test sample in the feature space. Unlike the traditional local learning and representation based approaches that are based on only one similarity measure, our individualized learning exploits three similarity measures, i.e., the kernel version of the Euclidean distance, the representation residual and cosine distance in the feature space. Secondly, although the determined learning area is a part of the entire training set, it is still high-dimensional and tends to be nonlinear separable. Since the kernel methods are suitable for dealing with the nonlinear separable data and can effectively avoid the small sample size problem [23], we use KFDA to build model for each test sample. And finally, we use a classifier such as the nearest neighbor classifier to classify the test sample. We refer to our proposed algorithm as individualized KFDA, i.e., IKFDA.

Compared with the conventional local learning and representation based approaches, the proposed IKFDA has the following nice properties. First of all, the proposed algorithm can build a desirable learning model for each test sample and can sufficiently learn the individual test samples. Second, it is clear that using multiple similarity measures can improve the robustness of determining the learning area for the test samples. However, as the conventional local learning and sparse representation approaches use only one similarity measure [24,25], they may not guarantee to obtain the appropriate training subset for the individual test samples. Third, when dealing with the large-scale and high-dimensional data sets, the conventional local learning and representation based approaches often need some preprocessing methods such as PCA to reduce the data dimensionality before building the learning model. On the contrary, IKFDA does not need the preprocessing methods of dimensionality reduction in the learning procedure in principle. Extensive experiments show that the IKFDA algorithm can obtain desirable classification results.

The rest of the paper is organized as follows: In Section 2, we describe the main idea of the individualized learning. Section 3 introduces a concrete individualized learning algorithm, i.e., IKFDA. Section 4 gives the experimental results and illustrates the effectiveness of the proposed algorithms. Section 5 offers our conclusions.

2. Main idea of the individualized learning

The main characteristic of the individualized learning is that it learns and classifies the test samples one by one. For a test sample, we first exploit the similarity measure to determine the learning area for this test sample. Then, we build a learning model within the learning area and extract the features of the test sample. Finally, we use a classifier, e.g., the nearest neighbor classifier, to classify the test sample. From the above individualized learning procedure, we observe that the individualized learning behaves locally (or is defined in a local manner). Thus, the individualized learning is consistent, i.e., for any distribution, it achieves the lowest possible expected loss as $L \to \infty$ where L is the number of the training samples [26]. In this sense, the individualized learning can work well without paying attention to the sample distribution. However, most commonized learning algorithms need to consider the sample distribution. For example, the well-known LDA algorithm performs well under the assumption that the data distribution is Gaussian. The general individualized learning framework is depicted in Fig. 1.

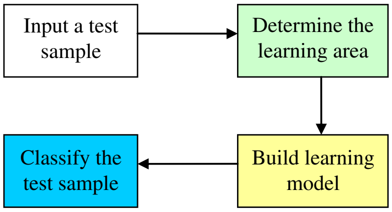


Fig. 1. The general individualized learning framework.

Actually, any algorithm can be viewed as one type of the individualized learning as long as it learns and classifies the test samples one by one. In the general individualized learning framework, if we employ only one similarity measure, e.g., the Euclidean distance often used in the traditional local algorithms, to select the training subset and build a learning model within this subset, the individualized learning is essentially a traditional local learning algorithm. On the other hand, when we use the representation residual as a similarity measure, the individualized learning can be converted into the representation classification method such as SRC. Based on the different learning settings, we can develop different concrete individualized learning algorithms. As discussed in Section 1, KFDA is suitable for building the learning model for the high-dimensional data sets. Since this work mainly focuses on how to learn the large-scale high-dimensional data sets, it is natural to adopt the KFDA to build the learning model in our work. In order to sufficiently learn the individual test samples, we need to determine the learning area by using the similarity measures for each test sample before building the learning model based on KFDA. Thus, we obtain a novel learning algorithm, IKFDA, which will be introduced in the following section.

3. Individualized KFDA (IKFDA)

In Section 2, we give the general individualized learning algorithm framework. In this section, we will propose a concrete individualized learning algorithm. The proposed algorithm contains two main steps. The first step is to use three similarity measures to determine the learning area for a test sample. The second step is to use the determined learning area to build the learning model. We first give the determination of the learning area in Section 3.1. Second, we build the KFDA learning model within the learning area in Section 3.2. Third, we give the complexity analysis in Section 3.3.

3.1. Learning area

As mentioned in Section 1, employing only one type of similarity measure may not guarantee to find an appropriate learning area for a test sample. In the determination of the learning area, we believe that the scheme using multiple similarity measures is more robust than that using only one similarity measure. Here, we use three similarity measures to improve the robustness of determining the learning area. Since KFDA is performed in the high dimensional feature space generated by one nonlinear mapping which is specified by applying some type of kernel functions, three similarity measures we used are also computed in this feature space, and they are the kernel version of the Euclidean distance, the representation residual and the cosine similarity measure (i.e., cosine distance) in the feature space.

Before determining the learning area for a test sample, we need to specify a value K that indicates the number of the samples that are most similar to the test sample in terms of each similarity measure. Suppose that the training set is denoted by $X=[x_1,x_2,\dots,x_L]$, where $x_i \in R^D (i=1,2,\dots,L)$ is the ith training sample and D is the sample dimensionality. $y \in R^D$ is a test sample. We use a nonlinear mapping ($\varphi : R^D \to F$ where F is the feature space), to map all the samples into the feature space. Then, we obtain the training set $\varphi(X)=[\varphi(x_1),\varphi(x_2),\dots,\varphi(x_L)]$, and the test sample $\varphi(y)$ in the feature space. In the following, we use three similarity measures to determine three similarity sets for the test sample, respectively. The first similarity measure is computed via the kernel version of Euclidean distance which is referred to as the kernelized Euclidean distance here.

3.1.1. Kernelized Euclidean distance

Given two samples in the feature space $\varphi(y)$ and $\varphi(x_i)$, where $\varphi(y)$ is a test sample and $\varphi(x_i)$ is a training sample. By using the kernel trick [27], the inner product $(\varphi(x_i),\varphi(x_j))$ between two samples $\varphi(x_i)$ and $\varphi(x_j)$ in the feature space can be computed by the kernel function $k(x_i,x_j)$. Then, the Euclidean distance between them is defined as

$$\begin{aligned} d &= \|\varphi(y)-\varphi(x_i)\|_2^2 = (\varphi(y)-\varphi(x_i))^T(\varphi(y)-\varphi(x_i)) \\ &= (\varphi(y),\varphi(y)) - 2(\varphi(x_i),\varphi(y)) + (\varphi(x_i),\varphi(x_i)) \\ &= k(y,y) - 2k(x_i,y) + k(x_i,x_i). \end{aligned} \tag{1}$$

Thus, we apply Eq. (1) to determine K nearest neighbors of the test sample $\varphi(y)$, and these neighbors consists of the first similarity set of $\varphi(y)$. This similarity set is denoted by $S_d=\{\varphi(x_d^1),\varphi(x_d^2),\dots,\varphi(x_d^K)\}$, where $\varphi(x_d^i) \in F (i=1,2,\dots,K)$ is the ith neighbor of the test sample $\varphi(y)$. The labels of the K nearest neighbors are $l_d^1, l_d^2, \dots, l_d^K$.

3.1.2. Cosine similarity measure in the feature space

We determine the second similarity set by employing the cosine similarity measure. In the feature space, the similarity between the training sample $\varphi(x_i)$ and the test sample $\varphi(y)$ can be measured by computing the cosine distance between these samples as follows:

$$\cos(\varphi(x_i),\varphi(y)) = \frac{(\varphi(x_i),\varphi(y))}{\|\varphi(x_i)\| \cdot \|\varphi(y)\|}. \tag{2}$$

By means of kernel trick

$$\cos(\varphi(x_i),\varphi(y)) = \frac{k(x_i,y)}{\sqrt{k(x_i,x_i)\cdot k(y,y)}}. \tag{3}$$

Note that if we use the Gaussian kernel function, then $k(x_i,x_i)=1$. Thus, Eq. (3) becomes

$$\cos(\varphi(x_i),\varphi(y)) = k(x_i,y). \tag{4}$$

We know that the larger cosine distance between the training sample and the test sample indicates that the training sample is more similar to the test sample. For a given test sample, we choose K training samples that correspond to the first K largest cosine distances. These training samples constitute the second similarity set $S_c=\{\varphi(x_c^1),\varphi(x_c^2),\dots,\varphi(x_c^K)\}$ and their labels are $l_c^1, l_c^2, \dots, l_c^K$.

3.1.3. Kernel representation residual

The third similarity set is determined by employing the kernel representation method proposed in [21,28,29]. Suppose that there is a vector $\beta=[b_1,b_2,\dots,b_L]^T$ satisfies the following equation:

$$\begin{aligned} \varphi(y) &= \varphi(X)\beta = [\varphi(x_1),\varphi(x_2),\dots,\varphi(x_L)]\beta \\ &= b_1\varphi(x_1) + b_2\varphi(x_2) + \cdots + b_L\varphi(x_L). \end{aligned} \tag{5}$$

Left dot-multiplying (5) with $\varphi(x_i)$, we can transform Eq. (5) into the following equation:

$$(\varphi(x_i)\cdot\varphi(y)) = b_1(\varphi(x_1)\cdot\varphi(x_i)) + \cdots + b_L(\varphi(x_L)\cdot\varphi(x_i)) \quad (i=1,\dots,L). \tag{6}$$

That is

$$k(x_i,y) = b_1 k(x_1,x_i) + \cdots + b_L k(x_L,x_i)(i=1,\dots,L). \tag{7}$$

The above equation can be rewritten as

$$K\beta = y_k, \tag{8}$$

where

$$K = \begin{pmatrix} k(x_1,x_1) & \cdots & k(x_L,x_1) \\ \vdots & \ddots & \vdots \\ k(x_1,x_L) & \cdots & k(x_L,x_L) \end{pmatrix}, \quad y_k = \begin{pmatrix} k(x_1,y) \\ \vdots \\ k(x_L,y) \end{pmatrix}.$$

If K is nonsingular, then we can solve Eq. (8) as follows:

$$\beta = K^{-1} y_k. \tag{9}$$

Otherwise, we have

$$\beta = (K + \varepsilon I)^{-1} y_k, \tag{10}$$

where ε is a small positive constant (in this work, ε is set to 0.01), and I is the identity matrix.

After obtaining β, we use the following formula to evaluate the similarity between the training sample $\varphi(x_i)$ and the testing sample $\varphi(y)$

$$\begin{aligned} re_i &= \|\varphi(y) - b_i\varphi(x_i)\|_2^2 \\ &= (\varphi(y) - b_i\varphi(x_i))^T(\varphi(y) - b_i\varphi(x_i)) \\ &= k(y,y) - 2b_i k(x_i,y) + b_i^2 k(x_i,x_i) \end{aligned} \tag{11}$$

Based on the above equation, we can use the training sample $\varphi(x_i)$ to represent the testing sample $\varphi(y)$, and re_i is referred to as the kernel representation residual. If a training sample has less residual of representing a test sample, then this training sample is more similar to the test sample. Thus, we can choose K training samples that correspond to the first K least kernel representation residuals, respectively. These samples constitute the third similarity set $S_r = \{\varphi(x_r^1), \varphi(x_r^2), ..., \varphi(x_r^K)\}$ and their labels are $l_r^1, l_r^2, ..., l_r^K$. From Eq. (11), the kernel representation residual can be viewed as an extension of the kernelized Euclidean distance. Actually, if b_i is equal to 1 in Eq. (11), the kernel representation residual reduces to the kernelized Euclidean distance.

3.1.4. *Determining the learning area*

For a test sample, we need to obtain a learning area for this sample after determining three similarity sets in the above subsections. To this end, we first measure the similarity between the test sample and each class that contains the samples in the similarity sets. The number of the total samples in three similarity sets is denoted as V ($V{=}3K$). Suppose the samples in the similarity sets belong to q classes: $\{C_1', C_2', ..., C_q'\}$. And among these sets, there are n_p samples belonging to Class $C_p'(p = 1, 2, ..., q)$. Then, the similarity between the test sample and Class C_p' is calculated as n_p/V. It is clear that large n_p/V indicates high similarity between the test sample and Class C_p'. Hence, we can obtain a similarity value set $S = \{S_1, S_2, ..., S_q\}$ in which the values are sorted in descending order. These values may be identical in some special case. In practice, we can choose the appropriate number of the samples in the similarity sets to avoid this case. We believe that the training samples associated with the smallest values in the value set S are usually not helpful to learn the test sample. In general, these samples may be noises or outliers within the data. Therefore, it is necessary to discard these samples from the similarity sets. Thus, the remaining training samples yield a new set, i.e., the learning area. It is used to build the learning model. We denote the samples in this learning area as $S_{IL} = [\varphi(x_{IL}^1), \varphi(x_{IL}^2), ..., \varphi(x_{IL}^M)]$ where M is the number of the samples in the learning area. And the labels of these determined samples are $l_{IL}^1, l_{IL}^2, ..., l_{IL}^M$. Note that among three above similarity measures used in the feature space, the first similarity measure can evaluate the correlation between the samples. Unlike the kernelized Euclidean distance, the cosine distance measure focuses on the difference on the orientations of sample vectors. The third measure evaluates the similarity between the samples from the viewpoint of the representation, and can be viewed as an extension of the kernelized Euclidean distance. These similarity measures can capture three types of information from the data set. Compared with each similarity set, the learning area S_{IL} integrated from three similarity sets can find more appropriate neighbors to sufficiently learn the test samples.

Fig. 2 gives an example of learning area. Fig. 2a shows a test sample that is from Class 35 on the ORL face data set which contains 40 persons (classes) and each person has 10 face images [30]. We use the first similarity measure, i.e., the kernelized Euclidean distance, to determine the first five nearest neighbors for the test sample. The determined neighbors are shown in Fig. 2b. From Fig. 2b, we can see that these nearest neighbors are not from Class 35. Therefore, if we use these neighbors to build the training model, it is clear that the model cannot correctly learn and classify the test image. In this case, we can say that the kernelized Euclidean distance is not a suitable similarity measure for this test image when we determine its five nearest neighbors. In other words, the kernelized Euclidean distance cannot find an appropriate training subset containing five nearest neighbors to learn this test sample. In this work, we use Gaussian kernel function (the Gaussian kernel parameter is set to 1.6e4) to determine the neighbors. Thus, the neighbors obtained by kernelized Euclidean distance are the same as ones obtained by cosine distance in the feature. Fig. 2c shows the similarity set containing five nearest neighbors obtained by using the kernel representation residual. We observe that the label of the test sample is included in the labels of these neighbors. Actually, the first two neighbors and the test sample are from the same class (Class 35) in Fig. 2c. Fig. 2d shows the final learning area. Also, the final learning area contains these two neighbors. Therefore, compared with the neighbors shown in Fig. 2b, the learning area is more suitable for learning the test sample in Fig. 2a.

3.2. *Learning model using KFDA*

In the real applications, the large scale and high dimensional data sets are usually highly nonlinear. Therefore, the samples in the learning area we determined in the previous subsection are generally nonlinear. Note that many traditional local learning algorithms and representation based approaches, e.g., SRC, often need some dimensionality reduction method such as PCA as a preprocessing procedure before learning. We believe that using this dimensionality reduction method before learning can loss the useful information for learning and classification. To address this problem, we adopt the kernel Fisher discriminant analysis (KFDA), which is very suitable for the nonlinear data sets, to build the learning model and classify the test sample. Indeed, KFDA does not need the dimensionality reduction as the preprocessing before learning in theory.

KFDA is the nonlinear case of the linear discriminant analysis [31]. The basic idea of KFDA is that the input data are mapped into a high dimensional feature space F by using a nonlinear mapping $\varphi : R^D \to F$. Then, we perform the linear discriminant analysis in this feature space [32]. After obtaining the learning area $S_{IL} = [\varphi(x_{IL}^1), \varphi(x_{IL}^2), ..., \varphi(x_{IL}^M)]$, we can build our learning model. The between-class and total scatter matrices of the learning area are denoted as S_b^φ and S_t^φ respectively. The matrices S_b^φ and S_t^φ are as follows:

$$S_b^\varphi = \sum_{i=1}^{l} n_i(\varphi(m_i) - \varphi(m))(\varphi(m_i) - \varphi(m))^T, \tag{12}$$

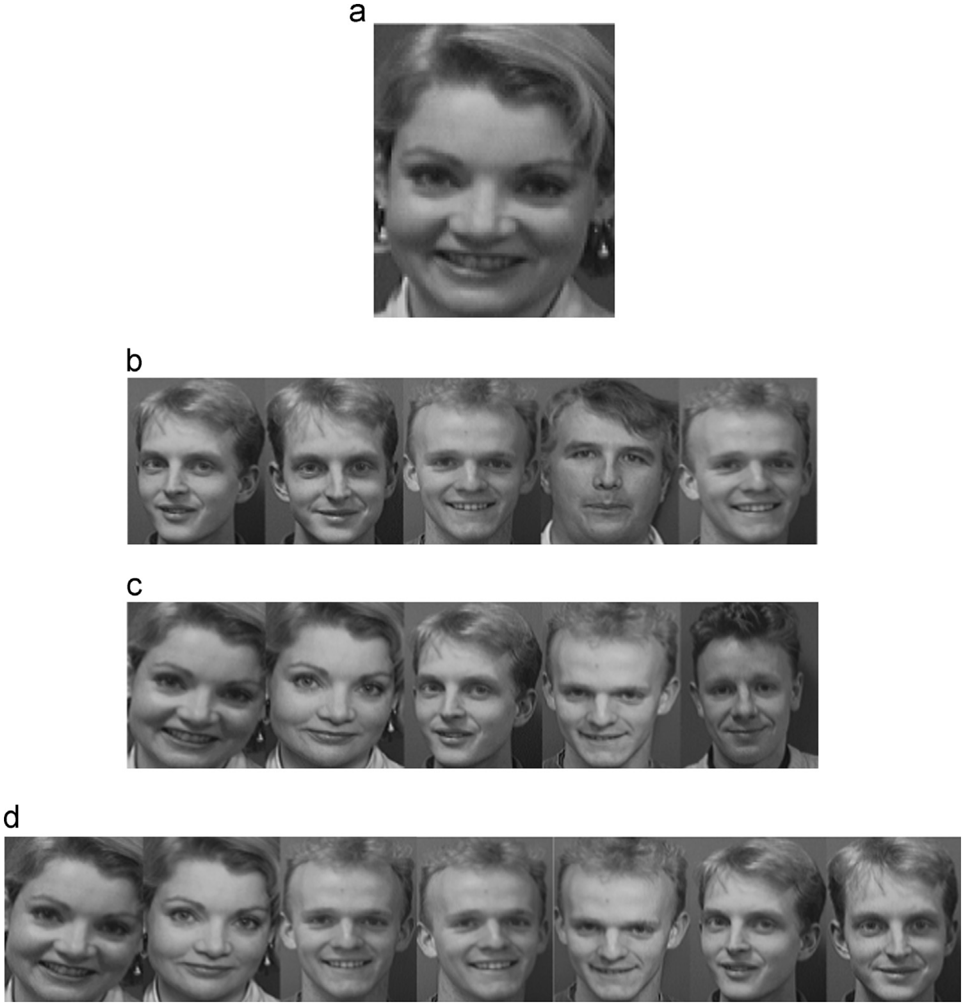


Fig. 2. A test sample and its similarity sets: (a) the test image, (b) the five nearest neighbors of the test image obtained by using the kernelized Euclidean distance, (c) the five nearest neighbors of the test image obtained by using the kernel representation residual and (d) final learning area of the test sample.

Table 1
Classification results on the AR data set.

Algorithms	N=3	N=4	N=5	N=6	N=7
1NN	48.67 ± 1.98	54.27 ± 1.48	60.12 ± 1.20	63.91 ± 1.02	67.04 ± 0.97
LDA	76.59 ± 1.51	82.97 ± 0.95	86.08 ± 1.20	88.80 ± 1.00	89.75 ± 0.87
KFDA	79.62 ± 1.97	85.11 ± 1.27	88.74 ± 0.86	91.64 ± 1.02	93.02 ± 0.96
SVM	60.21 ± 1.60	68.43 ± 1.52	74.89 ± 1.52	78.75 ± 0.97	81.27 ± 1.69
LRC	59.83 ± 1.32	68.29 ± 1.14	74.80 ± 1.32	79.56 ± 1.53	83.64 ± 0.96
SRC	80.35 ± 1.50	86.06 ± 0.55	89.96 ± 0.93	92.03 ± 0.57	93.92 ± 0.62
LSVM	60.14 ± 1.32	69.84 ± 1.08	77.21 ± 1.32	82.42 ± 1.09	85.81 ± 1.02
IKFDA	***83.60*** ± ***1.15***	***88.49*** ± ***1.12***	***91.47*** ± ***1.22***	***93.40*** ± ***0.56***	***94.83*** ± ***0.59***

$$S_t^{\varphi} = \sum_{j=1}^{M}(\varphi(x_{IL}^{j}) - \varphi(m))(\varphi(x_{IL}^{j}) - \varphi(m))^T. \tag{13}$$

where l is the number of the classes in the mapped learning area, n_i is the number of samples in the ith class such that $\sum_{i=1}^{l} n_i = M$, and two vectors $\varphi(m_i)$ and $\varphi(m)$ are the centroid of the ith class and the global centroid, respectively, in the learning area. The optimal projective vector β can be obtained via the following objective function [7]:

$$\beta_{opt} = \arg\max \frac{\beta^T S_b^{\varphi} \beta}{\beta^T S_t^{\varphi} \beta}. \tag{14}$$

According to [7], $\beta = \sum_{i=1}^{M} a_i \varphi(x_{IL}^{i})$, and (14) can be solved by the following eigen-problem:

$$GWG\alpha = \lambda GG\alpha. \tag{15}$$

where $\alpha = [a_1, a_2, ..., a_M]^T$, G is the kernel matrix in which each entry $G_{ij} = (\varphi(x_i), \varphi(x_j)) = k(x_i, x_j)$ and k is a kernel function, and W is as follows:

$$W_{ij} = \begin{cases} 1/n_k, & \text{if } x_i \text{and } x_j \text{ belong to the } k\text{th class} \\ 0, & \text{otherwise} \end{cases}$$

If we obtain the eigenvector α in (15), we can extract the features of the test sample $\varphi(y)$,

$$(\beta,\varphi(y)) = \sum_{i=1}^{M} a_i(\varphi(x_{IL}^i),\varphi(y)) = \sum_{i=1}^{M} a_i k(x_{IL}^i, y). \tag{16}$$

Similarly, we extract the features of training samples in the learning area. Thus, we can exploit a classifier to classify the test sample. Algorithm 1 gives our proposed algorithm.

It is clear that the IKFDA algorithm is an extension of the KFDA algorithm. KFDA is a special case of IKFDA. In fact, if the learning region of a test sample is the whole training set in IKFDA, the classification performance of IKFDA is theoretically equivalent to that of KFDA.

Algorithm 1. Individualized KFDA (IKFDA)

1. Input the training set: $x_i \in R^D (i=1,2,...,L)$, and a test sample $y \in R^D$.
2. From the training set, use three similarity measures to determine the learning area of the test sample y. The training samples in this area are: $S_{IL}=[\varphi(x_{IL}^1),\varphi(x_{IL}^2),...,\varphi(x_{IL}^M)]$;
3. Within the determined learning area, build the KFDA learning model and extract the features of the test sample by employing (12)–(16).
4. Classify the test sample using a classifier (the nearest neighbor classifier).

3.3. Complexity analysis

In this subsection, we first theoretically analyze the time complexity of our algorithm. For convenience, we give some notations. Suppose that L is the number of the total training samples, D is the data dimensionality which is usually very high, T is the number of the testing samples, and M is the number of the training samples in the learning area. The proposed algorithm contains two main steps. The first step is to determine the learning area for a test sample. This step needs to compute three similarity sets by adopting the kernelized Euclidean distance, cosine distance and representation residual in the feature space, respectively. The time complexities of them are $O(LD)$, $O(LD)$ and $O(L^3+L^2D)$, respectively. The second step mainly involves the kernel matrix construction and its eigen-decomposition. Their time complexities are $O(L^2D)$ and $O(L^3)$, respectively. Hence, the time complexity of the proposed algorithm is about $O(TL^2(L+D))$.

Since IKFDA learns and classifies the test samples one by one, it is slower than KFDA. Nevertheless, learning and classifying one test sample does not affect learning and classifying another test sample in IKFDA. That is, learning and classifying the test samples is parallely executed in IKFDA. If we perform IKFDA via parallel computation, the computational efficiency of IKFDA will be largely improved.

Second, we consider the space complexity. The space complexity of our algorithm is $O(MD+M^2)$, and that of the traditional KFDA is $O(LD+L^2)$. When the training set is very large-scale and the data dimensionality is very high, the traditional commonized learning algorithms such as LDA and KFDA might fail to learn if the main memory cannot load the whole training set. In this case, if we let the learning area of the test sample be a very small part of the training set, e.g.,$M=0.1{}^*L$ without significantly degrading the classification performance, the space complexity of our proposed algorithm is much lower than that of the traditional KFDA. In this sense, the proposed algorithm is more suitable for learning the very large-scale and high-dimensional data sets than the traditional KFDA. In addition, the proposed algorithm IKFDA can effectively improve the classification results of the traditional KFDA in many large-scale learning settings as demonstrated in the following experiments.

4. Experiments

In this section, we have conducted four experiments to evaluate the effectiveness of the proposed algorithm. Our experiments use four real-world data sets: the AR, YaleB, ORL and MNIST data sets. Among four data sets, the AR, YaleB and ORL are face data sets, and the MNIST is a handwritten digit data set. The first and second experiments are conducted on the AR and YaleB data sets, respectively. The third experiment is conducted on a new heteroscedastic data set which is combined by the AR and ORL data sets. The fourth experiment is conducted on the MNIST data set. Each sample in these data sets is a gray scale image with 256 gray levels per pixel. We compare our method with seven other state-of-the-art classification methods: KNN ($K=1$), LDA+NN (i.e., use the nearest neighbor classifier after LDA), KFDA+NN (use the nearest neighbor classifier after KFDA), SVM , linear regression classification (LRC) [33], SRC [17] and local SVM (LSVM) [16,34]. In the KNN, LDA+NN (denoted by LDA, for simplicity), KFDA+NN (denoted by KFDA) and our proposed algorithm, the nearest neighbor classifier is the Euclidean distance based on L2 norm.

In LDA and KFDA, the number of the transformation axes is $c-1$ where c is the number of the classes in the data sets. We use the Gaussian kernel function in KFDA, SVM implemented by using *LIBSVM* tool [35], and our algorithm. The optimal Gaussian kernel function parameters in KFDA, SVM and LSVM are obtained using the cross validation. For the SRC algorithm, the dimensionality of all images is first reduced to 300 by using PCA as a preprocessing procedure. Then, we perform SRC on the dimension-reduced data sets. There are two important parameters in our algorithm. The first parameter is the Gaussian kernel parameter. The second one is related to the learning area, denoted by R which indicates the ratio of the number of the training samples in each similarity set to the number of the total training samples. Also, the parameter R is determined by the cross validation on each data set. We report the best classification results of these algorithms in our experiments.

4.1. Experiment on the AR face data set

The first experiment is conducted on the AR face database. It contains over 4000 face images of 126 individuals, which include frontal views of faces with different facial expressions, lighting conditions, and occlusions [36,37]. We used the face images of 120 people and each people has 26 images. All the images are cropped and resized to a resolution of 50 × 40 pixels. We implemented our proposed algorithm and seven state-of-the-art classification algorithms mentioned above.

We randomly grouped the image samples of each individual into two parts. One part is used for training and the other part is used for testing. The number of training images that is chosen for each individual is 3, 4, 5, 6 and 7 which make up the five subsets of training data. As a result, the numbers of images in these five subsets are 360, 480, 600, 720 and 840, respectively. The Gaussian kernel parameter in KFDA is set to 0.0005* d where d is the average distance of two arbitrary samples in the subset of training data. For computational convenience, the Gaussian kernel parameter in our IKFDA algorithm is set to 0.001* d_1 where d_1 is the average distance of two arbitrary samples in the learning area of each individual test sample. It is worth noting that the Gaussian kernel parameter of IKFDA (i.e., 0.001* d_1) is not optimal for each test sample. Nevertheless, the classification performance of the IKFDA algorithm is still better than that of other classification algorithms as shown in Table 1. If we use the cross validation to select the optimal Gaussian kernel parameter in the learning area of each test sample, we can obtain the better classification performance. In the following experiments, our IKFDA algorithm uses the same scheme to determine the Gaussian kernel parameter for the purpose of computational simplicity. The parameter R in our algorithm is set to 0.3.

We randomly ran each algorithm 10 times. Table 1 shows the classification results which include the average classification accuracies and the standard deviations of the classification accuracies for each algorithm. In this table, N denotes the number of face images of each individual, and the bold italics highlight the best classification result on each subset of training data. According to Table 1, our proposed algorithm significantly outperforms the other state-of-the-art classification algorithms.

4.2. Experiment on the YaleB face data set

We conducted the second experiment on the YaleB face data set which contains 10 individuals with 5850 face images [38]. Each individual has 585 face images. The YaleB data set is a widely used illumination data set with images spanning a large range of possible illuminations [39]. In YaleB data set, each image is manually aligned and cropped, and the size of each image is 40 × 30. We randomly grouped the image samples of each individual into two parts. One part is used for training and the other part is used for testing. The number of training images that is chosen for each individual is 5 and 10 which make up two subsets of training data. The Gaussian kernel parameter in KFDA is set to 0.002* d where d is defined in the first experiment. For simplicity, the Gaussian kernel parameter in our algorithm is set to 0.002*d_1 where d_1 is also defined in the first experiment. The parameter R in our algorithm is set to 0.5.

We randomly ran each algorithm 10 times. Similar to Table 1, Table 2 shows the classification results of each algorithm and N denotes the number of face images of each individual, and the bold italics highlight the best classification result on each subset of training data in this table. From Table 2, we can see that our proposed algorithm outperforms the other state-of-the-art classification algorithms.

4.3. Experiment on the AR+ORL face data set

The third experiment is conducted on a new data set which is combined by the AR and ORL data sets (i.e., AR+ORL). We denote this new data set as the OA data set. The AR face database is the same as that used in the first experiment. The ORL face database contains 40 individuals with 400 face images. Each individual has 10 images. These images were captured at different times and have different variations including expression and facial details [40,41]. It is clear that the variations of the AR database and the ORL database are different. Hence, the new database, i.e., the OA database, is heteroscedastic. We will show later that our proposed algorithm is still suitable for learning this heteroscedastic data set and achieves the desirable classification results.

Table 2
Classification results on the YaleB data set.

Algorithms	N=5	N=10
1NN	87.02 ± 1.97	92.66 ± 1.72
LDA	95.81 ± 2.51	99.21 ± 0.68
KFDA	96.30 ± 2.37	99.31 ± 0.39
SVM	83.53 ± 4.75	94.31 ± 2.09
LRC	93.68 ± 1.45	98.67 ± 0.75
SRC	94.93 ± 2.14	98.09 ± 0.74
LSVM	90.56 ± 3.84	95.90 ± 1.60
IKFDA	***96.39 ± 2.21***	***99.35 ± 0.47***

In this experiment, all the images are cropped and resized to a resolution of 32 × 32 pixels. Similar to the first experiment, we randomly grouped the image samples of each individual into two parts. One part is used for training and the other part is used for testing. The number of training images that is chosen for each individual is 3, 4, 5, 6 and 7 which make up five subsets of training data. As a result, the numbers of images in the five subsets of training data are 480, 640, 800, 960 and 1120. The Gaussian kernel parameter in our algorithm is set to 0.002*d_1 where d_1 is also defined in the first experiment. The parameter R in our algorithm is set to 0.2.

We randomly ran each algorithm 10 times. Table 3 shows the classification results which include the average classification accuracies and the standard deviations of the classification accuracies for each algorithm. Also, N denotes the number of face images of each individual, and the bold italics highlight the best classification result on each subset of training data in this table. From Table 3, we can see that our proposed algorithm outperforms the other state-of-the-art classification algorithms.

4.4. Experiment on the MNIST data set

The fourth experiment is conducted on the MNIST data set [42] that is a handwritten digit data set with ten classes, i.e., 0,1,2,…,9 (each numeral corresponds to a class). For each numeral, the training set contains 6000 image samples, and the test set contains 1000 image samples. The size of each image is 28 × 28. We randomly selected the samples from the training set of each class and used them as the training samples in this experiment. The number of training samples that is chosen for each class is 5, 10, 15 and 20 which make up four subsets of training data. For each subset, we randomly selected the samples from the test set of each class and used them as the test samples in this experiment. The number of test samples that is chosen for each class is 300. Thus, we generate four test subsets respectively corresponding to four subsets of training data. Each test subset contains 3000 test samples. The Gaussian kernel parameter in KFDA is set to 0.0002*d where d is the average distance of two arbitrary samples in the subset of training data. Similar to the previous experiments, the Gaussian kernel parameter in our algorithm is set to 0.002*d_1 where d_1 is the average distance of two arbitrary samples in the learning area. The parameter R in our algorithm is set to 0.1. We randomly ran each algorithm 10 times. Table 4 reports the best classification results on four subsets of training data for each algorithm. In this table, N denotes the number of training samples of each class, and the bold italics highlight the best classification result on each subset. According to Table 4, IKFDA achieves the highest classification accuracies among eight classification algorithms when N=5 and 10. In other cases, i.e., N=15 and 20, the proposed algorithm can achieve similar or better performance in comparison with the other state-of-the-art classification algorithms. Notice that the classification results of the LRC algorithm are undesirable in the first three experiments. From our four experiments, the proposed algorithm can achieve the most stable and desirable classification results among these classification algorithms. As a whole, the proposed algorithm is the best algorithm among these state-of-the-art classification algorithms in terms of the classification result.

4.5. Relationship between the R and the classification performance

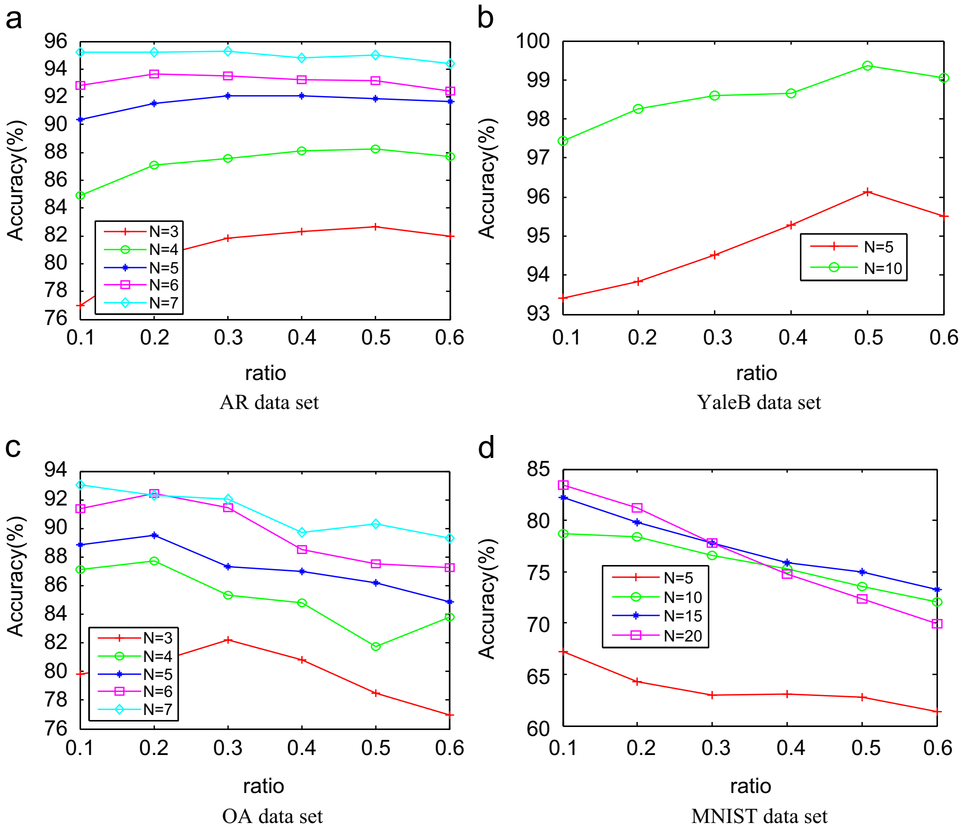
In this subsection, we will investigate the relationship between the parameter related to the learning area R and the classification

Table 3
Classification results on the OA(AR+ORL) data set.

Algorithms	$N=3$	$N=4$	$N=5$	$N=6$	$N=7$
1NN	52.0 ± 0.65	57.16 ± 1.04	61.5 ± 1.08	65.63 ± 0.74	67.94 ± 0.84
LDA	72.91 ± 1.45	78.05 ± 0.91	79.66 ± 1.23	78.71 ± 0.99	77.75 ± 1.21
KFDA	77.41 ± 1.14	83.98 ± 0.84	87.02 ± 3.62	89.93 ± 2.75	92.43 ± 0.76
SVM	61.33 ± 1.20	68.82 ± 1.25	74.90 ± 2.07	77.52 ± 1.35	80.64 ± 1.24
LRC	62.07 ± 1.87	69.31 ± 0.79	75.59 ± 1.24	80.07 ± 1.60	84.36 ± 1.17
SRC	79.27 ± 1.30	85.14 ± 1.15	88.78 ± 0.84	91.53 ± 0.58	92.93 ± 0.77
LSVM	61.97 ± 1.06	69.91 ± 0.82	75.82 ± 1.63	79.75 ± 0.99	81.18 ± 1.35
IKFDA	***81.12*** ± ***1.13***	***86.28*** ± ***0.86***	***89.52*** ± ***1.26***	***91.93*** ± ***0.63***	***93.15*** ± ***0.56***

Table 4
Classification results on the MNIST data set.

Algorithms	$N=5$	$N=10$	$N=15$	$N=20$
INN	59.66 ± 3.02	68.71 ± 2.24	74.59 ± 1.12	76.81 ± 1.30
LDA	61.16 ± 2.63	65.33 ± 2.42	66.49 ± 2.24	64.69 ± 1.55
KFDA	63.93 ± 3.19	73.62 ± 1.98	78.33 ± 1.51	79.16 ± 1.34
SVM	67.62 ± 2.66	77.88 ± 2.68	82.85 ± 1.40	84.71 ± 0.66
LRC	67.91 ± 2.59	78.13 ± 2.73	***84.47*** ± ***1.12***	***85.96*** ± ***0.91***
SRC	60.56 ± 2.80	69.04 ± 2.43	73.97 ± 0.79	74.89 ± 0.76
LSVM	67.83 ± 2.78	78.12 ± 2.72	83.29 ± 1.39	85.07 ± 0.63
IKFDA	***68.12*** ± ***2.82***	***78.17*** ± ***2.42***	82.47 ± 1.05	83.34 ± 0.67



AR data set

YaleB data set

OA data set

MNIST data set

Fig. 3. The relationship between the ratio R and the classification accuracy of IKFDA on four data sets: (a) AR data set; (b) YaleB data set; (c) OA data set; and (d) MNIST data set.

performance on the AR, YaleB, OA and MNIST data sets. In the experiment, we randomly ran the IKFDA algorithm one time on each subset of training data. Fig. 3 shows the relationship between the ratio R and the classification accuracy of the IKFDA algorithm. For each subset, we have plotted the classification accuracy curve versus the variation of the values of the R (R=0.1, 0.2, 0.3, 0.4, 0.5 and 0.6) in this figure. From Fig. 3, we can see that the small values of the R usually lead to better classification results on most of training subsets. As shown in Fig. 3, the appropriate ratios on the four data sets range from 0.1 to 0.5. We can easily choose the best ratio which leads to the highest classification accuracy from the ratio value set {0.1, 0.2, 0.3 0.4 0.5}. Thus, it is not difficult to effectively select the appropriate parameter R for the IKFDA algorithm.

5. Conclusions and future work

In this paper, we have introduced a general novel learning framework, i.e., the individualized learning that learns and classifies the test samples one by one. The individualized learning algorithms are consistent and can perform well without considering the data distribution in general. In order to deal well with the large-scale and high-dimensional data sets, we have proposed a concrete individualized learning algorithm, i.e., the individualized KFDA (IKFDA) algorithm. The proposed algorithm exploits multiple similarity measures to determine the learning area and builds the KFDA learning model within this learning area. The extensive experiments show that the IKFDA algorithm can achieve desirable classification results and outperform other popular state-of-the-art classification algorithms as a whole.

The individualized learning framework aims to sufficiently learn and classify each test sample. Its concrete learning form indeed relies on the learning settings. Different learning settings can lead to different concrete individualized learning algorithms. In the future work, we will combine the individualized learning framework with the other learning algorithms such as Adaboost algorithm to sufficiently learn both the training samples and test samples.

Conflict of interest

There is no conflict of interests.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (NSFC) under grants Nos. 61472138, 61263032, 71262011, 61271385, and 61362031, and the Shenzhen Municipal Science and Technology Innovation Council No. JCYJ20150330155220591, as well as Science and Technology Foundation of Jiangxi Transportation Department of China (2015D0066).

References

[1] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, et al., Fisher discriminant analysis with kernels, in: Neural Networks for Signal Processing IX, 1999, pp. 41–48.
[2] G. Wang, N. Shi, Y. Shu, D. Liu, Embedded manifold-based kernel Fisher discriminant analysis for face recognition, Neural Process. Lett. 43 (1) (2016) 1–16.
[3] M.A. Tahir, J. Kittler, A. Bouridane, Multi-label classification using stacked spectral kernel discriminant analysis, Neurocomputing 171 (2016) 127–137.
[4] H.-K. Min, Y. Hou, S. Park, I. Song, A computationally efficient scheme for feature extraction with kernel discriminant analysis, Pattern Recognit. 50 (2016) 45–55.
[5] X. Han, L. Clemmensen, Regularized generalized eigen-decomposition with applications to sparse supervised feature extraction and sparse discriminant analysis, Pattern Recognit. 49 (2016) 43–54.
[6] Y. Xu, D. Zhang, Z. Jin, M. Li, et al., A fast kernel-based nonlinear discriminant analysis for multi-class problems, Pattern Recognit. 39 (6) (2006) 1026–1033.
[7] D. Cai, X. He, J. Han, Speed up kernel discriminant analysis, VLDB J. 20 (1) (2011) 21–33.
[8] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (7) (1997) 711–720.
[9] Z. Fan, Y. Xu, W. Zuo, J. Yang, et al., Modified principal component analysis: an integration of multiple similarity subspace models, IEEE Trans. Neural Netw. Learn. Syst. 25 (8) (2014) 1538–1552.
[10] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cognit. Neurosci. 3 (1) (1991) 71–86.
[11] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.
[12] C.W. Hsu, C.J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 415–425.
[13] X. Jiang, Linear subspace learning-based dimensionality reduction, IEEE Signal Process. Mag. 28 (2) (2011) 16–26.
[14] N. Segata, E. Blanzieri, Fast and scalable local kernel machines, J. Mach. Learn. Res. 11 (6) (2010) 1883–1926.
[15] M. Sugiyama, T. Ide, S. Nakajima, J. Sese, Semi-supervised local Fisher discriminant analysis for dimensionality reduction, Mach. Learn. 78 (1) (2010) 35–61.
[16] H. Zhang, A.C. Berg, M. Maire, J. Malik, SVM-KNN: Discriminative nearest neighbor classification for visual category recognition, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 2126–2136.
[17] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, et al., Robust face recognition via sparse representation, IEEE Trans. pattern Anal. Mach. Intell. 31 (2) (2009) 210–227.
[18] J. Yang, L. Zhang, Y. Xu, J.-y Yang, Beyond sparsity: the role of L1-optimizer in pattern classification, Pattern Recognit. 45 (3) (2012) 1104–1118.
[19] Z. Fan, M. Ni, Q. Zhu, C. Sun, et al., L0-norm sparse representation based on modified genetic algorithm for face recognition, J. Vis. Commun. Image Represent. 28 (2015) 15–20.
[20] S. Gao, I.W.-H. Tsang, L.-T. Chia, Sparse representation with kernels, IEEE Trans. Image Process. 22 (2) (2013) 423–434.
[21] Y. Xu, D. Zhang, J. Yang, J.Y. Yang, et al., A two-phase test sample sparse representation method for use with face recognition, IEEE Trans. Circuits Syst. Video Technol. 21 (9) (2011) 1255–1262.
[22] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: which helps face recognition? in: 2011 IEEE International Conference on Computer Vision (ICCV), 2011, pp. 471–478.
[23] L.-F. Chen, H.-Y.M. Liao, M.-T. Ko, J.-C. Lin, et al., A new LDA-based face recognition system which can solve the small sample size problem, Pattern Recognit. 33 (10) (2000) 1713–1726.
[24] Mien Van, H.-J. Kang, Bearing defect classification based on individual wavelet local fisher discriminant analysis with particle swarm optimization, IEEE Trans. Ind. Inform. 12 (1) (2016) 124–135.
[25] W. Liu, Z. Yu, L. Lu, Y. Wen, et al., KCRC-LCD: discriminative kernel collaborative representation with locality constrained dictionary for visual categorization, Pattern Recognit. 48 (10) (2015) 3076–3092.
[26] A. Zakai, Y. Ritov, Consistency and localizability, J. Mach. Learn. Res. 10 (4) (2009) 827–856.
[27] B. Scholkopf, The kernel trick for distances, Adv. Neural Inf. Process. Syst. (2001) 301–307.
[28] Q. Zhu, Y. Xu, J. Wang, Z. Fan, Kernel based sparse representation for face recognition, in: Proceedings of the 21st International Conference on Pattern Recognition (ICPR), 2012, pp. 1703–1706.
[29] Y. Xu, Z. Fan, Q. Zhu, Feature space-based human face image representation and recognition, Opt. Eng. 51 (1) (2012) 017205–1-017205-7.
[30] J. Yang, D. Zhang, A.F. Frangi, J.-y Yang, Two-dimensional PCA: a new approach to appearance-based face representation and recognition, IEEE Trans. Pattern Anal. Mach. Intell. 26 (1) (2004) 131–137.
[31] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Comput. 12 (10) (2000) 2385–2404.
[32] Y. Xu, J. Yang, J. Lu, D. Yu, An efficient renovation on kernel Fisher discriminant analysis and face recognition experiments, Pattern Recognit. 37 (10) (2004) 2091–2094.
[33] I. Naseem, R. Togneri, M. Bennamoun, Linear regression for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (11) (2010) 2106–2112.
[34] E.K. Garcia, S. Feldman, M.R. Gupta, S. Srivastava, Completely lazy learning, IEEE Trans. Knowl. Data Eng. 22 (9) (2010) 1274–1285.
[35] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 27.
[36] A.M. Martinez, A.C. Kak, PCA versus lDA, IEEE Trans. Pattern Anal. Mach. Intell. 23 (2) (2001) 228–233.
[37] J. Yang, D. Zhang, J.-y Yang, B. Niu, Globally maximizing, locally minimizing: unsupervised discriminant projection with applications to face and palm biometrics, IEEE Trans. Pattern Anal. Mach. Intell. 29 (4) (2007) 650–664.
[38] R. Basri, T. Hassner, L. Zelnik-Manor, Approximate nearest subspace search, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2) (2010) 266–278.
[39] J.R. Beveridge, B.A. Draper, J.M. Chang, M. Kirby, et al., Principal angles separate subject illumination spaces in YDB and CMU-PIE, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 351–356.
[40] W.H. Yang, D.Q. Dai, Two-dimensional maximum margin feature extraction for face recognition, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 39 (4) (2009) 1002–1012.
[41] Y. Xu, Z. Zhang, G. Lu, J. Yang, Approximately symmetrical face images for image preprocessing in face recognition and sparse representation based classification, Pattern Recognit. (2016).
[42] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86 (11) (1998) 2278–2324.

Zizhu Fan received the Ph.D. degree in Computer Science and Technology at the Shenzhen Graduate School, Harbin Institute of Technology (HIT), China, in 2014. Now he is an associate professor at the School of Basic Science in East China Jiaotong University. His current interests include pattern recognition and image processing. He has published more than 20 journal papers.

Yong Xu received his Ph.D. degree in Pattern Recognition and Intelligence System at the Nanjing University of Science and Technology (NUST), Nanjing, China, in 2005. Now he is an associate professor at Shenzhen Graduate School, HIT. His interests include pattern recognition and machine learning. He has published more than 40 scientific papers.

Ming Ni received the Ph.D. degree in Information Management from the University of Shanghai for Science and Technology, Shanghai, China, in 2006. Now he is an associate professor at the School of Basic Science in East China Jiaotong University. His current interests include intelligence information processing. He has published more than 30 journal papers.

Xiaozhao Fang received the M.S. degree in Computer Science from the Guangdong University of Technology, Guangzhou, China, in 2008. He is currently pursuing the Ph.D. degree in Computer Science and Technology at the Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. He has published more than 7 journal papers. His current research interests include pattern recognition and machine learning.

David Zhang received the Ph.D. degree in Computer Science from the HIT, Harbin, China, in 1985. In 1994 he received his second Ph.D. in Electrical and Computer Engineering from the University of Waterloo, Ontario, Canada. Currently, he is a Head of Department of Computing, and a Chair Professor at the Hong Kong Polytechnic University. He is the Founder and Editor-in-Chief, International Journal of Image and Graphics (IJIG); Associate Editor of more than ten international journals including IEEE Transactions and Pattern Recognition; and the author of more than 10 books and 200 journal papers. Professor Zhang is a Fellow of both IEEE and IAPR.