Signal Processing 90 (2010) 2333–2343

Contents lists available at ScienceDirect

Signal Processing

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

Kernel-based nonlinear discriminant analysis using minimum squared errors criterion for multiclass and undersampled problems ☆

Wen-Jun Zeng [a,*], Xi-Lin Li [b], Xian-Da Zhang [c], En Cheng [a]

[a] Key Laboratory of Underwater Acoustic Communication and Marine Information Technology of the Ministry of Education, Xiamen University, Xiamen 361005, China

[b] Department of Computer Science and Electrical Engineering, University of Maryland Baltimore County, Baltimore, MD 21250, USA

[c] Department of Automation, Tsinghua University, Beijing 100084, China

ARTICLE INFO

Article history:
Received 7 October 2008
Received in revised form
30 March 2009
Accepted 1 June 2009
Available online 6 June 2009

Keywords:
Dimensionality reduction
Discriminant analysis
Kernel methods
Minimum squared errors
Undersampled problem

ABSTRACT

It is well known that there exist two fundamental limitations in the linear discriminant analysis (LDA). One is that it cannot be applied when the within-class scatter matrix is singular, which is caused by the undersampled problem. The other is that it lacks the capability to capture the nonlinearly clustered structure of the data due to its linear nature. In this paper, a new kernel-based nonlinear discriminant analysis algorithm using minimum squared errors criterion (KDA-MSE) is proposed to solve these two problems. After mapping the original data into a higher-dimensional feature space using kernel function, the MSE criterion is used as the discriminant rule and the corresponding dimension reducing transformation is derived. Since the MSE solution does not require the scatter matrices to be nonsingular, the proposed KDA-MSE algorithm is applicable to the undersampled problem. Moreover, the new KDA-MSE algorithm can be applied to multiclass problem, whereas the existing MSE-based kernel discriminant methods are limited to handle twoclass data only. Extensive experiments, including object recognition and face recognition on three benchmark databases, are performed and the results demonstrate that our algorithm is competitive in comparison with other kernel-based discriminant techniques in terms of recognition accuracy.

1. Introduction

Linear discriminant analysis (LDA) is a typical method for feature extraction and dimensionality reduction. LDA finds a linear transformation which maximizes the between-class scatter and minimizes the within-class scatter to achieve the maximum class separability. The transformation of LDA can be calculated by performing the generalized eigenvalue decomposition (GEVD) of the within-class scatter matrix and the between-class scatter matrix [1,2]. LDA has been successfully applied in a variety of pattern classification and machine learning fields, such as face recognition, text categorization, and DNA analysis, etc. However, there exist two main drawbacks of the conventional LDA. The first shortcoming is that the conventional LDA requires within-class scatter matrix must be nonsingular and the second one is that as a linear method, LDA lacks the capability to capture the nonlinearly clustered structure of the data.

All of the scatter matrices are singular is a common case since in many practical applications, the number of samples is smaller than the data dimension. In such case, the total-scatter matrix is singular and the conventional LDA is no longer applicable, which is referred to as the

☆ This work was supported in part by the National Natural Science Foundation of China under Grant 60672046 and 60675002, the Key Project of Fujian Province Science and Technology Foundation under Grant 2008H0036, and the Specialized Research Fund for the Doctorol Program of Higher Education.

* Corresponding author. Tel.: +86 592 2580081; fax: +86 592 2580038.
E-mail address: cengwj06@mails.tsinghua.edu.cn (W.-J. Zeng).

0165-1684/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2009.06.002

undersampled problem or *small sample size problem*. To overcome the limitation of conventional LDA, a variety of techniques have been proposed, e.g., PCA+LDA [3], regularized LDA [4], penalized LDA [5], null space-based LDA [6], direct LDA [7], LDA/GSVD [8], MMC [9], ULDA and OLDA [10,11]. In [11], a detailed review of LDA for undersampled problem is given.

An interesting and promising strategy to solve the undersampled problem is casting the LDA problem as a minimum squared errors (MSE) problem by considering the class label as the output. It is well known that for *twoclass* problem, LDA is equivalent to the MSE solution with a specific class label as the output [2]. Recently, the relationship between LDA and the MSE solution for *multiclass* and undersampled problems has been developed [12,13]. By exploiting such relationship, LDA can be performed through the MSE procedure without carrying out GEVD of the scatter matrices. Since there is no requirement on the nonsingularity of the scatter matrices for the MSE problem, naturally, the undersampled problem can be solved.

In many applications, the distribution of the data is complex and highly nonlinear. In such case, the performance of LDA degrades since it cannot capture the nonlinearly clustered structure of the data. In order to make LDA effective to nonlinearly distributed data, we need to conduct the nonlinear extension of LDA by kernel trick. The kernel methods are originally introduced in support vector machines (SVMs) [14]. The key idea of kernel methods is to map the original data to a higher-dimensional feature space where the inner products can be computed by a kernel function without knowing the nonlinear mapping function explicitly [14,15]. The pioneering kernel-based feature extractor is the kernel principal components analysis (KPCA) [16], which is a nonlinear extension of the well known PCA. Like PCA, the transformation (projection) of KPCA aims to preserve the maximum energy in the reduced space and it does not consider to keep or enhance the between-classes discriminant power. The kernel Fisher discriminant (KFD) [17,18] first extends LDA to nonlinear case by kernel function, however, KFD can only handle twoclass data. The generalized discriminant analysis (GDA) [19] is applicable to multiclass problem. Nevertheless, the theoretical development of GDA assumes that the kernel matrix of the centered data must be nonsingular. Such an assumption is violated since centering in feature space makes the kernel matrix singular, which results in performance degrading [20]. Moreover, GDA requires to perform eigenvalue decompositions twice so its computational burden is heavy.

Note that the dimension of the feature space is often much higher than that of the original data space; therefore the undersampled problem becomes more severe. To address the undersampled problem, several kernel discriminant analysis (KDA) approaches have been proposed by generalizing the corresponding LDA techniques which can be applied to undersampled situations. By extending the idea of PCA+LDA, KPCA plus LDA is proposed in [22] and it is a two-phase method. In conjunction with the direct LDA and the kernel method, the kernel direct discriminant analysis (KDDA) algorithm is developed [21]. KDDA can be considered as a one-stage method since there is no separate PCA step. The kernel maximum margin criterion (KMMC) [9] and kernel scatter-difference-based discriminant analysis (KSDA) [23] are based on the difference of between-class scatter matrix minus the weighted within-class scatter matrix. However, the optimal weight of the within-class scatter matrix is difficult to determine. The kernel uncorrelated discriminant analysis (KUDA) [24] and kernel orthogonal discriminant analysis (KODA) [24] are nonlinear extensions of ULDA and OLDA, respectively, and the two methods perform well in nonlinear and undersampled case. But KUDA requires to carry out the singular value decompositions (SVD) twice and KODA three times, which results in expensive computation. Based on generalization singular value decomposition (GSVD) [25] and kernel trick, the LDA/GSVD is extended to the kernel version: KDA/GSVD in [20]. It is an effective approach for dealing with undersampled and nonlinear problems. The main shortcoming of KDA/GSVD is the high computation cost of GSVD.

In this paper, motivated by the success of the kernel method in dealing with nonlinearly distributed data and the applicability of the MSE solution for undersampled problem, a new kernel-based nonlinear discriminant analysis algorithm using MSE criterion (KDA-MSE) for multiclass problem is first presented. In KDA-MSE, we first map the original data into a higher-dimensional feature space using kernel function. Then in the feature space, the MSE criterion is used as the discriminant rule and the corresponding dimension reducing transformation is derived. The MSE-based kernel discriminant analysis methods have been discussed in [26–28]. Nevertheless, they are only applicable to twoclass problem. In addition, the methods in [26–28] use regularization to handle the undesampled problem, which brings a troublesome problem that how to choose the optimal regularization parameter. Being different from the approaches in [26–28], the proposed KDA-MSE algorithm can be applied to multiclass problem and no additional regularization is required.

The remainder of this paper is organized as follows. In Section 2, the conventional LDA is reviewed, and LDA by MSE formulation is introduced. In Section 3, we present the new kernel discriminant analysis algorithm based on MSE criterion, KDA-MSE. Extensive experiments, including object recognition and face recognition, are conducted to compare the performances of KDA-MSE with other kernel-based discriminant analysis algorithms as well as LDA-MSE in Section 4. Finally, this paper is concluded in Section 5.

2. LDA by MSE formulation

In this section, we first review the conventional LDA and point out its limitations. Then we introduce how to formulate LDA as an MSE problem by multivariate linear regression model in multiclass and undersampled cases.

2.1. Conventional LDA

Consider a dataset containing n samples which belongs to c classes

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n], \tag{1}$$

where $\mathbf{x}_i \in \mathbb{R}^m$ is an m-dimensional vector. Denote $y_i \in \{1, 2, \ldots, c\}$ the class label of the i-th sample. Without loss of generality, partition $\mathbf{X}$ into c classes as

$$\mathbf{X} = [\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(c)}], \tag{2}$$

where $\mathbf{X}^{(i)} \in \mathbb{R}^{m \times n_i}$, n_i is the number of samples of the i-th class, and naturally $\sum_{i=1}^{c} n_i = n$. Let $\mathbf{x}_k^{(i)}$ represent the k-th sample of the i-th class to emphasize the class index. Therefore $\mathbf{X}^{(i)}$ can be written as

$$\mathbf{X}^{(i)} = [\mathbf{x}_1^{(i)}, \mathbf{x}_2^{(i)}, \ldots, \mathbf{x}_{n_i}^{(i)}]. \tag{3}$$

The between-class scatter matrix $\mathbf{S}_b$, within-class scatter matrix $\mathbf{S}_w$, and total-scatter matrix $\mathbf{S}_t$ are defined as

$$\mathbf{S}_b = \frac{1}{n}\sum_{i=1}^{c} n_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T, \tag{4}$$

$$\mathbf{S}_w = \frac{1}{n}\sum_{i=1}^{c}\sum_{k=1}^{n_i}(\mathbf{x}_k^{(i)} - \mathbf{m}_i)(\mathbf{x}_k^{(i)} - \mathbf{m}_i)^T, \tag{5}$$

$$\mathbf{S}_t = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T, \tag{6}$$

where $\mathbf{m}_i = (1/n_i)\sum_{k=1}^{n_i}\mathbf{x}_k^{(i)}$ is the centroid of the i-th class, $\mathbf{m} = (1/n)\sum_{i=1}^{n}\mathbf{x}_i$ is the global centroid, and superscript T denotes transpose. It is clear that $\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w$. The three scatter matrices can be expressed as

$$\mathbf{S}_b = \mathbf{H}_b\mathbf{H}_b^T, \quad \mathbf{S}_w = \mathbf{H}_w\mathbf{H}_w^T, \quad \mathbf{S}_t = \mathbf{H}_t\mathbf{H}_t^T, \tag{7}$$

where

$$\mathbf{H}_b = \left[\sqrt{\frac{n_1}{n}}(\mathbf{m}_1 - \mathbf{m}), \ldots, \sqrt{\frac{n_c}{n}}(\mathbf{m}_c - \mathbf{m})\right], \tag{8}$$

$$\mathbf{H}_w = \frac{1}{\sqrt{n}}\left[\mathbf{X}^{(1)} - \mathbf{m}_1\mathbf{e}_1^T, \ldots, \mathbf{X}^{(c)} - \mathbf{m}_c\mathbf{e}_c^T\right], \tag{9}$$

$$\mathbf{H}_t = \frac{1}{\sqrt{n}}(\mathbf{X} - \mathbf{m}\mathbf{e}^T) \tag{10}$$

and $\mathbf{e}_i$ is an $n_i \times 1$ vector and $\mathbf{e}$ is an $n \times 1$ vector with all elements being ones.

By maximizing the between-class scatter and minimizing the within-class scatter, the optimal linear transform matrix $\mathbf{W}$ for LDA can be obtained from [1]

$$\mathbf{W}_{\mathrm{LDA}} = \underset{\mathbf{W}}{\operatorname{argmax}}\{\mathrm{tr}((\mathbf{W}^T\mathbf{S}_t\mathbf{W})^{-1}(\mathbf{W}^T\mathbf{S}_b\mathbf{W}))\}, \tag{11}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix. It is well known that the optimization problem in (11) is equivalent to solving the generalized eigenvalue problem $\mathbf{S}_b\mathbf{w} = \lambda\mathbf{S}_t\mathbf{w}$ and the columns of $\mathbf{W}_{\mathrm{LDA}}$ are given by the generalized eigenvectors corresponding to the $c-1$ largest nonzero eigenvalues. When $\mathbf{S}_t$ is nonsingular, one can calculate $\mathbf{W}_{\mathrm{LDA}}$ by applying eigenvalue decomposition on matrix $\mathbf{S}_t^{-1}\mathbf{S}_b$. We refer to this method as conventional LDA. In twoclass problem (i.e., $c = 2$), the conventional LDA is referred to as Fisher discriminant analysis (FDA) since Fisher firstly proposed the idea of LDA for binary class classification [29]. Note that the reduced dimension of conventional LDA is at most $c-1$ since $\mathrm{rank}(\mathbf{S}_b) \le c-1$.

When the number of samples is smaller than the data dimension, the total-scatter matrix $\mathbf{S}_t$ is singular and the conventional LDA is no longer applicable. In order to solve the small sample size problem, LDA using MSE formulation is adopted.

2.2. LDA using MSE criterion

The MSE solution finds a linear discriminant function that minimizes the squared errors. It has been shown that in twoclass problem, FDA is equivalent to the MSE solution with a specific class label as the output [2]. In multiclass cases, a linear discriminant function for each class i is

$$f_i(\mathbf{x}) = b_i + \mathbf{w}_i^T\mathbf{x}, \quad i = 1, \ldots, c, \tag{12}$$

where $\mathbf{w}_i$ is the weight vector and b_i is the bias of the linear model. For a given sample $\mathbf{x}_j$ ($1 \le j \le n$), the output of the linear discriminant function is specified as

$$b_i + \mathbf{w}_i^T\mathbf{x}_j = g_{ji}. \tag{13}$$

Clearly, the specified output g_{ji} should be related to the class label for classification purpose. Different choices of the specified output will lead to different discriminant function. Consider the case where the specified outputs are centered, i.e.,

$$\frac{1}{n}\sum_{j=1}^{n} g_{ji} = 0, \quad i = 1, \ldots, c. \tag{14}$$

The centered sample data are denoted as

$$\tilde{\mathbf{X}} = [\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_n], \tag{15}$$

where

$$\tilde{\mathbf{x}}_j = \mathbf{x}_j - \mathbf{m}. \tag{16}$$

When both the samples and specified outputs are centered, the bias b_i becomes zero. It follows that

$$\mathbf{w}_i^T\tilde{\mathbf{x}}_j = g_{ji}. \tag{17}$$

For $1 \le j \le n$ and $1 \le i \le c$, the squared error cost function is expressed as

$$J = \left\| \begin{bmatrix} \tilde{\mathbf{x}}_1^T \\ \vdots \\ \tilde{\mathbf{x}}_n^T \end{bmatrix} [\mathbf{w}_1, \ldots, \mathbf{w}_c] - \begin{bmatrix} g_{11} & \cdots & g_{1c} \\ \vdots & & \vdots \\ g_{n1} & \cdots & g_{nc} \end{bmatrix} \right\|_{\mathrm{F}}^2, \tag{18}$$

where $\|\cdot\|_{\mathrm{F}}^2$ is the Frobenius norm. By denoting

$$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_c], \quad \mathbf{G} = \begin{bmatrix} g_{11} & \cdots & g_{1c} \\ \vdots & & \vdots \\ g_{n1} & \cdots & g_{nc} \end{bmatrix} \tag{19}$$

the cost function in Eq. (18) can be written as

$$J(\mathbf{W}) = \|\tilde{\mathbf{X}}^T\mathbf{W} - \mathbf{G}\|_{\mathrm{F}}^2 = \mathrm{tr}(\mathbf{W}^T\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\mathbf{W} - \mathbf{W}^T\tilde{\mathbf{X}}\mathbf{G} - \mathbf{G}^T\tilde{\mathbf{X}}^T\mathbf{W} + \mathbf{G}^T\mathbf{G}). \tag{20}$$

The MSE solution of the linear regression model is given by

$$\frac{\partial J(\mathbf{W})}{\partial \mathbf{W}} = 2(\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T\mathbf{W} - \tilde{\mathbf{X}}\mathbf{G}) = \mathbf{0}, \tag{21}$$

which leads to

$$\mathbf{W}_{\mathrm{MSE}} = (\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T)^{\dagger}\tilde{\mathbf{X}}\mathbf{G} = (n\mathbf{S}_t)^{\dagger}\tilde{\mathbf{X}}\mathbf{G}, \tag{22}$$

where superscript † denotes Moore–Penrose pseudoinverse.

Since matrix $\mathbf{G}$ is composed of the specified outputs, which are related to the class labels, we refer to $\mathbf{G}$ as *class indicator matrix*. There exist several methods for setting the specified outputs according to the class labels. An interesting choice of setting g_{ji} given by [13] is

$$g_{ji} = [\mathbf{G}]_{ji} = \begin{cases} \sqrt{\dfrac{n}{n_i}} - \sqrt{\dfrac{n_i}{n}} & \text{if } \tilde{\mathbf{x}}_j \in \text{ class } i, \\ -\sqrt{\dfrac{n_i}{n}} & \text{otherwise,} \end{cases} \tag{23}$$

which leads to

$$\tilde{\mathbf{X}}\mathbf{G} = n\mathbf{H}_b. \tag{24}$$

Using such class indicator matrix, the MSE solution can be obtained by

$$\mathbf{W}_{\mathrm{MSE}} = \mathbf{S}_t^{\dagger}\mathbf{H}_b. \tag{25}$$

When the sample size is large enough, $\mathbf{S}_t$ is nonsingular and $\mathbf{S}_t^{\dagger}$ equals $\mathbf{S}_t^{-1}$. When there appears the undersampled problem, $\mathbf{S}_t$ is singular and the Moore–Penrose pseudoinverse should be adopted. Using the pseudoinverse-based MSE solution, the small sample size problem is successfully circumvented.

In [12,13], the relationships between LDA and the MSE solution are analyzed in multiclass cases. It is revealed that the MSE solution is related to the dimension reducing matrix of LDA. Therefore we can use the $m \times c$ matrix $\mathbf{W}_{\mathrm{MSE}}$ to perform dimensionality reduction instead of the generalized eigenvalue decomposition required by LDA. The matrix $\mathbf{W}_{\mathrm{MSE}}$ is referred to as LDA-MSE transformation and for any data point $\mathbf{z}$, implementing LDA-MSE transformation leads to a c-dimensional representation with enhanced discriminatory capability,

$$\mathbf{W}_{\mathrm{MSE}}^T\mathbf{z} = \mathbf{H}_b^T\mathbf{S}_t^{\dagger}\mathbf{z}. \tag{26}$$

3. Kernel discriminant analysis by MSE criterion

As mentioned above, it is difficult for linear discriminant methods to describe the complex and nonlinear distribution of the data. Therefore the nonlinear extension of LDA-MSE by exploiting kernel function is required. In this section, the kernel discriminant analysis algorithm using MSE criterion for multiclass data is first derived.

3.1. Kernel functions and kernel matrix

Assume that a nonlinear mapping function $\Phi(\cdot)$ maps the input space to a higher-dimensional feature space:

$$\mathbf{x} \in \mathbb{R}^m \to \Phi(\mathbf{x}) \in \mathbb{R}^M. \tag{27}$$

Note that the dimension of feature space is often much larger than that of the original data space, i.e., $M \gg m$. Moreover, M can even be infinity. Therefore the undersampled problem becomes more severe after nonlinear mapping. The dataset in the feature space can be written as

$$\Phi(\mathbf{X}) = [\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \ldots, \Phi(\mathbf{x}_n)], \tag{28}$$

$$\Phi(\mathbf{X}^{(i)}) = [\Phi(\mathbf{x}_1^{(i)}), \Phi(\mathbf{x}_2^{(i)}), \ldots, \Phi(\mathbf{x}_{n_i}^{(i)})], \quad i = 1, 2, \ldots, c. \tag{29}$$

Since for any kernel function κ satisfying Mercer's condition [14], there exists a mapping Φ such that

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \langle \Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2) \rangle = \Phi^T(\mathbf{x}_1)\Phi(\mathbf{x}_2), \tag{30}$$

where $\langle , \rangle$ denotes the inner product. One can achieve the nonlinear mapping by computing inner products in feature space by means of kernel functions in input space. Therefore it is not necessary to carry out the nonlinear mapping explicitly. The two most widely used kernel functions are Gaussian kernel and polynomial kernel. Gaussian kernel uses an inner product

$$\kappa(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(\frac{-\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2}\right), \tag{31}$$

where σ is the kernel parameter to be adjusted.

Given the dataset $\mathbf{X}$ containing n samples, the $n \times n$ kernel matrix $\mathbf{K}$ is defined as

$$\mathbf{K} = \Phi^T(\mathbf{X})\Phi(\mathbf{X}). \tag{32}$$

Clearly, the (i,j)-th entry of $\mathbf{K}$ is

$$[\mathbf{K}]_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j), \quad 1 \le i, j \le n. \tag{33}$$

For convenience of the following derivation, we need to center the data in the high-dimensional feature space. The data points

$$\tilde{\Phi}(\mathbf{x}_i) = \Phi(\mathbf{x}_i) - \mathbf{m}^{\Phi} \tag{34}$$

are centered. Here $\mathbf{m}^{\Phi} = (1/n)\sum_{i=1}^{n} \Phi(\mathbf{x}_i)$ is the global centroid in the feature space. Throughout the paper, $\tilde{\Phi}$ denotes the centering operation after nonlinear mapping. The centered dataset in feature space can be represented as

$$\tilde{\Phi}(\mathbf{X}) = [\tilde{\Phi}(\mathbf{x}_1), \tilde{\Phi}(\mathbf{x}_2), \ldots, \tilde{\Phi}(\mathbf{x}_n)], \tag{35}$$

$$\tilde{\Phi}(\mathbf{X}^{(i)}) = [\tilde{\Phi}(\mathbf{x}_1^{(i)}), \tilde{\Phi}(\mathbf{x}_2^{(i)}), \ldots, \tilde{\Phi}(\mathbf{x}_{n_i}^{(i)})], \quad i = 1, 2, \ldots, c. \tag{36}$$

Similarly, the kernel matrix of the centered data is defined as

$$\tilde{\mathbf{K}} = \tilde{\Phi}^T(\mathbf{X})\tilde{\Phi}(\mathbf{X}). \tag{37}$$

It has been shown that the centered kernel matrix can be calculated by (see [16])

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_{n\times n}\mathbf{K} - \mathbf{K}\mathbf{1}_{n\times n} + \mathbf{1}_{n\times n}\mathbf{K}\mathbf{1}_{n\times n}, \tag{38}$$

where all the elements of $n \times n$ matrix $\mathbf{1}_{n\times n}$ are equal to $1/n$. Generally speaking, the ranks of matrices $\mathbf{K}$ and $\tilde{\mathbf{K}}$ are

$$\mathrm{rank}(\mathbf{K}) = n, \quad \mathrm{rank}(\tilde{\mathbf{K}}) = n - 1. \tag{39}$$

Eq. (39) means that centering in feature space makes the centered kernel matrix $\tilde{\mathbf{K}}$ singular. In the theoretical development of GDA, it assumes that $\tilde{\mathbf{K}}$ must be nonsingular [19]. Nonetheless, such assumption does not hold, which results in the performance degradation of GDA.

3.2. KDA-MSE algorithm

According to Eq. (25), by applying LDA-MSE in the high-dimensional feature space, one can obtain the transformation of kernel discriminant analysis based on minimum squared errors:

$$\mathbf{W}^{\Phi} = (\mathbf{S}_t^{\Phi})^{\dagger}\mathbf{H}_b^{\Phi}, \tag{40}$$

where $\mathbf{S}_t^{\Phi}$ is the total-scatter matrix of the data in high-dimensional feature space given by

$$\mathbf{S}_t^{\Phi} = \frac{1}{n}\sum_{i=1}^{n}\tilde{\Phi}(\mathbf{x}_i)\tilde{\Phi}^T(\mathbf{x}_i) = \frac{1}{n}\tilde{\Phi}(\mathbf{X})\tilde{\Phi}^T(\mathbf{X}) \tag{41}$$

and

$$\mathbf{H}_b^{\Phi}(\mathbf{H}_b^{\Phi})^T = \mathbf{S}_b^{\Phi}, \tag{42}$$

$$\mathbf{H}_b^{\Phi} = \left[\sqrt{\frac{n_1}{n}}\mathbf{m}_1^{\tilde{\Phi}}, \sqrt{\frac{n_2}{n}}\mathbf{m}_2^{\tilde{\Phi}}, \ldots, \sqrt{\frac{n_c}{n}}\mathbf{m}_c^{\tilde{\Phi}}\right], \tag{43}$$

where $\mathbf{m}_i^{\tilde{\Phi}} = (1/n_i)\sum_{i=1}^{n_i}\tilde{\Phi}(\mathbf{x}_1^{(i)})$ is the centroid of the i-th class in the feature space. However, it is difficult to calculate the KDA-MSE transformation matrix $\mathbf{W}^{\Phi}$ directly according to Eq. (40) since the nonlinear mapping is usually unknown. Even if the nonlinear mapping is explicitly given, it is unpractical to calculate $\mathbf{S}_t^{\Phi}$ and $\mathbf{H}_b^{\Phi}$ directly because of the very high dimensionality in the feature space (it is also referred to as curse of dimensionality). To circumvent the curse of dimensionality, we develop an effective algorithm to implement KDA-MSE based on kernel tricks in the following.

Since the rank of symmetric positive semidefinite matrix $\mathbf{S}_t^{\Phi}$ is $r = \mathrm{rank}(\mathbf{S}_t^{\Phi}) = n-1$, the number of nonzero eigenvalues of $\mathbf{S}_t^{\Phi}$ is r and the other eigenvalues equal zeros. The eigenvalue decomposition of $\mathbf{S}_t^{\Phi}$ can be expressed as

$$\mathbf{S}_t^{\Phi} = [\mathbf{U}_s \ \mathbf{U}_n]\begin{bmatrix}\boldsymbol{\Sigma}_s & \\ & \mathbf{0}\end{bmatrix}\begin{bmatrix}\mathbf{U}_s^T \\ \mathbf{U}_n^T\end{bmatrix} = \mathbf{U}_s\boldsymbol{\Sigma}_s\mathbf{U}_s^T, \tag{44}$$

where

$$\boldsymbol{\Sigma}_s = \mathrm{diag}(\lambda_1^{\Phi}, \lambda_2^{\Phi}, \ldots, \lambda_r^{\Phi}) \tag{45}$$

is a diagonal matrix containing the r nonzero eigenvalues in descending order and

$$\mathbf{U}_s = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_r] \tag{46}$$

contains the r corresponding orthonormal eigenvectors, i.e.,

$$\mathbf{S}_t^{\Phi}\mathbf{u}_i = \lambda_i^{\Phi}\mathbf{u}_i, \quad i = 1, \ldots, r. \tag{47}$$

The columns of $\mathbf{U}_n$ are the orthonormal eigenvectors that correspond to the zero eigenvalues.

Since $\mathbf{U}_s$ is orthonormal, the Moore–Penrose pseudoinverse of $\mathbf{S}_t^{\Phi}$ can be calculated as [25]

$$(\mathbf{S}_t^{\Phi})^{\dagger} = \mathbf{U}_s\boldsymbol{\Sigma}_s^{-1}\mathbf{U}_s^T, \tag{48}$$

which leads to

$$\mathbf{W}^{\Phi} = \mathbf{U}_s\boldsymbol{\Sigma}_s^{-1}\mathbf{U}_s^T\mathbf{H}_b^{\Phi}. \tag{49}$$

For any input data point $\mathbf{z}$, our final purpose is to calculate the projection onto the range space of the dimension reducing matrix $\mathbf{W}^{\Phi}$, i.e.,

$$\mathbf{z}_{\mathrm{KDA}} = (\mathbf{W}^{\Phi})^T\tilde{\Phi}(\mathbf{z}) = (\mathbf{H}_b^{\Phi})^T\mathbf{U}_s\boldsymbol{\Sigma}_s^{-1}\mathbf{U}_s^T\tilde{\Phi}(\mathbf{z}). \tag{50}$$

As mentioned above, directly solving the eigenvalue problem $\mathbf{S}_t^{\Phi}\mathbf{u}_i = \lambda_i^{\Phi}\mathbf{u}_i$ is not possible due to the curse of dimensionality. Note that the eigenvectors corresponding the nonzero eigenvalues must lie in the space spanned by $\tilde{\Phi}(\mathbf{x}_1), \ldots, \tilde{\Phi}(\mathbf{x}_n)$. Hence there exists a set of coefficients $\boldsymbol{\beta}_i = [\beta_{i1}, \ldots, \beta_{in}]^T$ such that $\mathbf{u}_i$ can be expanded as

$$\mathbf{u}_i = \sum_{k=1}^{n}\beta_{ik}\tilde{\Phi}(\mathbf{x}_k) = \tilde{\Phi}(\mathbf{X})\boldsymbol{\beta}_i, \quad i = 1, \ldots, r. \tag{51}$$

Define a matrix

$$\mathbf{B} = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_r] \tag{52}$$

then $\mathbf{U}_s$ can be written as

$$\mathbf{U}_s = \tilde{\Phi}(\mathbf{X})\mathbf{B}. \tag{53}$$

By substituting $\mathbf{S}_t^{\Phi} = (1/n)\tilde{\Phi}(\mathbf{X})\tilde{\Phi}^T(\mathbf{X})$ and $\mathbf{u}_i = \tilde{\Phi}(\mathbf{X})\boldsymbol{\beta}_i$ into Eq. (47), we get

$$\frac{1}{n}\tilde{\Phi}(\mathbf{X})\tilde{\Phi}^T(\mathbf{X})\tilde{\Phi}(\mathbf{X})\boldsymbol{\beta}_i = \lambda_i^{\Phi}\tilde{\Phi}(\mathbf{X})\boldsymbol{\beta}_i. \tag{54}$$

Premultiplying both sides of (54) by $n\tilde{\Phi}^T(\mathbf{X})$ results in

$$\tilde{\Phi}^T(\mathbf{X})\tilde{\Phi}(\mathbf{X})\tilde{\Phi}^T(\mathbf{X})\tilde{\Phi}(\mathbf{X})\boldsymbol{\beta}_i = n\lambda_i^{\Phi}\tilde{\Phi}^T(\mathbf{X})\tilde{\Phi}(\mathbf{X})\boldsymbol{\beta}_i. \tag{55}$$

By noting that $\tilde{\mathbf{K}} = \tilde{\Phi}^T(\mathbf{X})\tilde{\Phi}(\mathbf{X})$, Eq. (55) can be written as

$$\tilde{\mathbf{K}}^2\boldsymbol{\beta}_i = n\lambda_i^{\Phi}\tilde{\mathbf{K}}\boldsymbol{\beta}_i, \quad i = 1, \ldots, r. \tag{56}$$

Finding the eigenvectors associated with *nonzero* eigenvalues of (56) is equivalent to solving the eigenvalue problem

$$\tilde{\mathbf{K}}\boldsymbol{\beta}_i = \lambda_i^K\boldsymbol{\beta}_i, \quad i = 1, \ldots, r \tag{57}$$

for nonzero eigenvalues. The nonzero eigenvalues of $\tilde{\mathbf{K}}$ and $\mathbf{S}_t^{\Phi}$ have the following relationship

$$\lambda_i^K = n\lambda_i^{\Phi}. \tag{58}$$

Define a diagonal matrix

$$\boldsymbol{\Sigma}_K = \mathrm{diag}(\lambda_1^K, \lambda_2^K, \ldots, \lambda_r^K), \tag{59}$$

whose diagonal entries are the eigenvalues of $\tilde{\mathbf{K}}$. Clearly, we have

$$\boldsymbol{\Sigma}_K = n\boldsymbol{\Sigma}_s. \tag{60}$$

Note that there exits a constraint to the norm of the coefficient vectors $\boldsymbol{\beta}_i$ since it is required that the corresponding vectors $\mathbf{u}_i$ should be normalized in

the high-dimensional space, i.e., $\|\mathbf{u}_i\|^2 = 1$, which leads to

$$\|\mathbf{u}_i\|^2 = \mathbf{u}_i^T\mathbf{u}_i = \boldsymbol{\beta}_i^T\tilde{\Phi}^T(\mathbf{X})\tilde{\Phi}(\mathbf{X})\boldsymbol{\beta}_i = \boldsymbol{\beta}_i^T\tilde{\mathbf{K}}\boldsymbol{\beta}_i = \lambda_i^K\|\boldsymbol{\beta}_i\|^2 = 1. \tag{61}$$

Therefore the norms of $\boldsymbol{\beta}_i$ should satisfy

$$\|\boldsymbol{\beta}_i\| = \frac{1}{\sqrt{\lambda_i^K}}, \quad i = 1,\ldots,r, \tag{62}$$

which means that the columns of $\mathbf{B}$ are only *orthogonal* but not *orthonormal*.

By substituting (53) and (60) into (50), we obtain

$$\mathbf{z}_{\mathrm{KDA}} = n(\mathbf{H}_b^{\Phi})^T\tilde{\Phi}(\mathbf{X})\mathbf{B}\boldsymbol{\Sigma}_K^{-1}\mathbf{B}^T\tilde{\Phi}^T(\mathbf{X})\tilde{\Phi}(\mathbf{z}). \tag{63}$$

Since $\mathbf{B}$ and $\boldsymbol{\Sigma}_K$ can be obtained from the eigenvectors and eigenvalues of the centered kernel matrix $\tilde{\mathbf{K}}$, the remaining problems are how to calculate $(\mathbf{H}_b^{\Phi})^T\tilde{\Phi}(\mathbf{X})$ and $\tilde{\Phi}^T(\mathbf{X})\tilde{\Phi}(\mathbf{z})$ by using kernel functions only. It can be proved that (see Appendix A)

$$n(\mathbf{H}_b^{\Phi})^T\tilde{\Phi}(\mathbf{X}) = \mathbf{E}\tilde{\mathbf{K}}, \tag{64}$$

where $\mathbf{E}$ is a $c \times n$ block diagonal matrix

$$\mathbf{E} = \sqrt{n}\begin{bmatrix} \frac{1}{\sqrt{n_1}}\mathbf{e}_1^T & & & \\ & \frac{1}{\sqrt{n_2}}\mathbf{e}_2^T & & \\ & & \ddots & \\ & & & \frac{1}{\sqrt{n_c}}\mathbf{e}_c^T \end{bmatrix}, \tag{65}$$

where $\mathbf{e}_i^T = [1,\ldots,1]$ is a $1 \times n_i$ row vector. The derivation of Eqs. (64) and (65) can be found in Appendix A.

The $n \times 1$ vector

$$\tilde{\mathbf{k}}_z := \tilde{\Phi}^T(\mathbf{X})\tilde{\Phi}(\mathbf{z}) \tag{66}$$

can be calculated as

$$\tilde{\mathbf{k}}_z = \mathbf{k}_z - \mathbf{1}_{n\times n}\mathbf{k}_z - \mathbf{K}\mathbf{1}_{n\times 1} + \mathbf{1}_{n\times n}\mathbf{K}\mathbf{1}_{n\times 1} \tag{67}$$

where all the elements of $n \times n$ matrix $\mathbf{1}_{n\times n}$ and $n \times 1$ vector $\mathbf{1}_{n\times 1}$ are equal to $1/n$, and the $n \times 1$ vector $\mathbf{k}_z$ is given by

$$\mathbf{k}_z = [\kappa(\mathbf{x}_1,\mathbf{z}), \kappa(\mathbf{x}_2,\mathbf{z}), \ldots, \kappa(\mathbf{x}_n,\mathbf{z})]^T. \tag{68}$$

The proof procedure of (67) is given by Appendix B. By substituting (64) and (66) into (63), we obtain the low-dimensional representation $\mathbf{z}_{\mathrm{KDA}}$ using the KDA-MSE dimension reducing transformation

$$\mathbf{z}_{\mathrm{KDA}} = \mathbf{E}\tilde{\mathbf{K}}\mathbf{B}\boldsymbol{\Sigma}_K^{-1}\mathbf{B}^T\tilde{\mathbf{k}}_z. \tag{69}$$

Note that the columns of $\mathbf{B}$ are only orthogonal but not orthonormal; therefore $\mathbf{B}\boldsymbol{\Sigma}_K^{-1}\mathbf{B}^T \neq \tilde{\mathbf{K}}^{\dagger}$. It is expected that the low-dimensional representation $\mathbf{z}_{\mathrm{KDA}}$ has enhanced discriminatory power. Different from other kernel discriminant analysis methods, the reduced dimension of KDA-MSE is c other than $c-1$.

3.3. Summarization of KDA-MSE

The detailed steps for implementing the KDA-MSE algorithm are summarized as follows.

KDA-MSE algorithm:

- *Input*: Training dataset $\mathbf{X}$ and the class labels, input data point $\mathbf{z}$.
- *Output*: The low-dimensional representation $\mathbf{z}_{\mathrm{KDA}}$.
- *Algorithm*:
  (1) Calculate kernel matrix $\mathbf{K}$ using (33).
  (2) Calculate centered kernel matrix $\tilde{\mathbf{K}}$ by (38).
  (3) Compute r nonzero eigenvalues $\{\lambda_i^K\}_{i=1}^r$ and the corresponding eigenvectors $\boldsymbol{\beta}_i$ of $\tilde{\mathbf{K}}$.
  (4) Scale each eigenvector $\boldsymbol{\beta}_i$ using (62).
  (5) Construct $\mathbf{B}$, $\boldsymbol{\Sigma}_K$ and $\mathbf{E}$ using (52), (59) and (65), respectively.
  (6) Use (67) and (68) to compute $\tilde{\mathbf{k}}_z$.
  (7) Obtain the low-dimensional representation $\mathbf{z}_{\mathrm{KDA}}$ according to (69).

The leading computation cost of the KDA-MSE algorithm is calculating the r nonzero eigenvalues and the corresponding eigenvectors of the $n \times n$ kernel matrix $\tilde{\mathbf{K}}$ with computational complexity being $\mathcal{O}(n^2r)$. Note that our KDA-MSE algorithm only needs to carry out the eigenvalue decomposition (EVD) *once*, whereas the GDA, KDDA and KUDA algorithms need to perform the EVD operations *twice*. Moreover, KODA needs to calculate EVD three times. Therefore the calculation amount of KDA-MSE is only half about of those of GDA, KDDA and KUDA, and only one-third of that of KODA.

4. Experimental results

In this section, extensive experiments are carried out to demonstrate that the proposed KDA-MSE is an effective nonlinear dimensionality reduction method.

4.1. Object recognition: comparing KDA-MSE with LDA-MSE

In the first experiment, we show that the nonlinear KDA-MSE is easier to capture the nonlinearly cluster structure and achieve better performance than LDA-MSE when the distribution of data is nonlinear and complex. We choose the COIL-20 database to demonstrate this advantage of our nonlinear kernel method.

The COIL-20 database [30] consisting of 1440 gray-scale images of 20 objects is used in the first experiment. The objects were placed on a motorized turntable against a black background. The turntable was rotated through 360° to vary object pose with respect to a fixed camera. Images of the objects were taken at pose intervals of 5°. This corresponds to 72 images per object. The images are downsampled to 32×32 for computation efficiency. Fig. 1 shows some images of three different objects.

In the experiment, the training set and testing set are obtained by randomly splitting the dataset. For each object, a random subset with p images is taken with labels to form the training set, and the rest of the database is considered to be the testing set. The values of p range from 2 to 20, i.e., $2 \leq p \leq 20$.

For each value of p, 50 independent runs are performed to obtain the average recognition accuracy. In each run, the Gaussian kernel function is adopted and the kernel

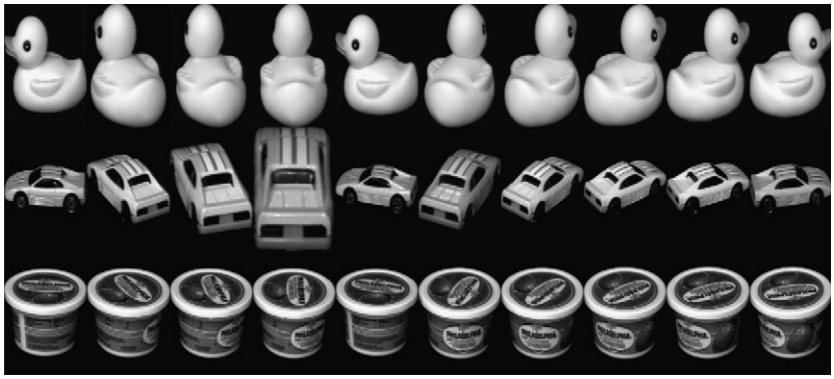

Fig. 1. Some sample images from the COIL-20 database.

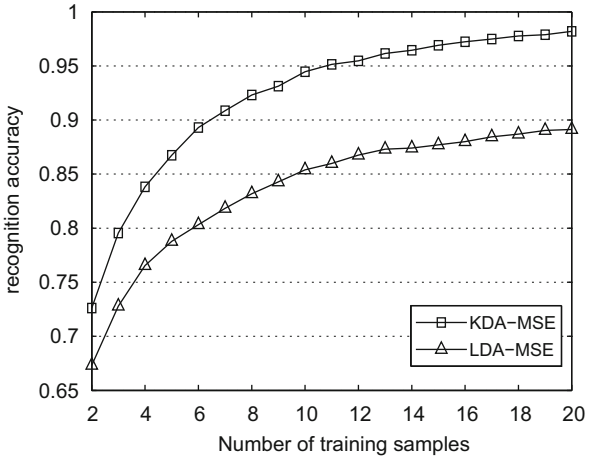

Fig. 2. Average recognition accuracies on COIL-20 database.

parameter σ is equal to the average of pairwise Euclidean distances in the training data, i.e.,

$$\sigma = \frac{1}{n(n-1)/2} \sum_{i=1}^{n} \sum_{j=i+1}^{n} \|\mathbf{x}_i - \mathbf{x}_j\|. \tag{70}$$

After dimension reducing, the nearest-neighbor (NN) classifier with Euclidean metric is adopted in the reduced dimensional space.

Fig. 2 illustrates the average recognition accuracies of KDA-MSE and LDA-MSE on COIL-20 database. Clearly, the average accuracy by using KDA-MSE is much higher than LDA-MSE. The images of the COIL-20 objects are collected under various viewpoints, and such multiview property results in a highly nonlinear and complex distribution of the data. As a linear method, LDA-MSE is not capable to describe the structure of nonlinearly separable datasets and fails to deliver good performance. However, KDA-MSE can solve this inherent nonlinear problem and achieve much higher classification accuracy.

4.2. Face recognition: comparing KDA-MSE with other kernel methods

We compare the performance of KDA-MSE with several kernel discriminant methods: KPCA [16], GDA [19], KUDA and KODA [24] for face recognition applications. Three benchmark face database: ORL face database [31], Yale face database [32] and UMIST face database [33] are used to perform the face recognition experiments.

The ORL face database consists of 10 different images of each of 40 distinct subjects for a total of 400 images. For some subjects, the images are taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images are taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). The resolution of the original images is 112×92 with 256 gray levels. We downsample the original images to 32×32 for computation efficiency. Fig. 3 displays 10 images of one subject.

The Yale face database contains 165 grayscale images of 15 individuals. There are 11 images per subject, one per different facial expression (normal, happy, sad, sleepy, surprised, and wink), lighting condition (left-light, center-light, right-light), and with/without glasses. The images are downsampled to 32×32 again for computation efficiency. Fig. 4 shows 10 images of one individual.

The UMIST face database consists of 564 images of 20 individuals covering a range of poses from profile to frontal views. The number of images of each subject vary from 19 to 36 and each image has 256 gray levels and of size 92×112, which is downsampled to 23×28 in our experiment. The 10 images of one subject are shown in Fig. 5.

In the face recognition experiments, for each individual, a random subset of p images is used for training, and the rest are used for testing. We set $2 \le p \le 9$ for the ORL database, $2 \le p \le 10$ for the Yale database, and $2 \le p \le 12$ for the UMIST database, respectively.

For each value of p, 50 independent trials are performed to obtain the average recognition accuracy. In each trial, the Gaussian kernel function is adopted and the kernel parameter σ is selected according to Eq. (70). KPCA uses the eigenvectors associated with the largest eigenvalues that preserve 95% of the variance. After dimension reducing, we still adopt the NN classifier in the reduced dimensional space.

Figs. 6, 7 and 8 illustrate the average recognition accuracies on ORL database, Yale database and UMIST

Fig. 3. Samples from the ORL face database.

Fig. 4. Samples from the Yale face database.

Fig. 5. Samples from the UMIST face database.

database, respectively. Clearly, KDA-MSE is superior to KPCA, GDA and KUDA, no matter which dataset is used or no matter whether the number of training samples is large or small. The proposed KDA-MSE is better than KODA for ORL face database. But KODA outperforms KDA-MSE a bit for Yale and UMIST face database. Generally, it can be concluded that KODA and KDA-MSE deliver the similar performance in terms of recognition accuracy. However, since KODA needs to implement the SVD three times, its calculation amount is approximately triple of that of KDA-MSE.

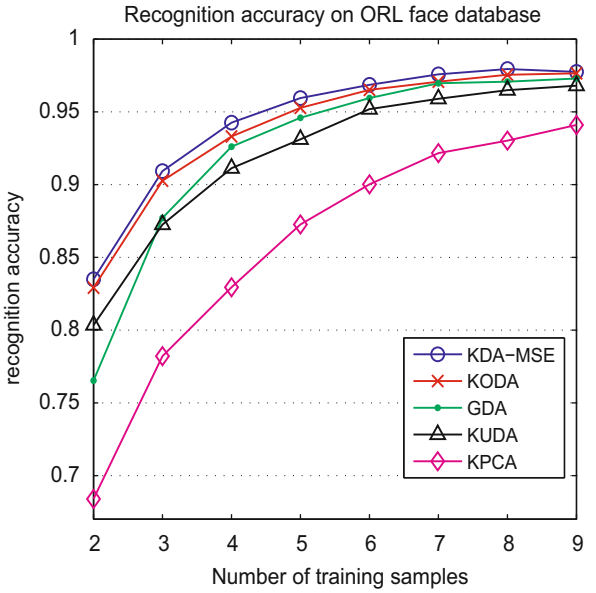

Fig. 6. Average recognition accuracies on ORL face database.

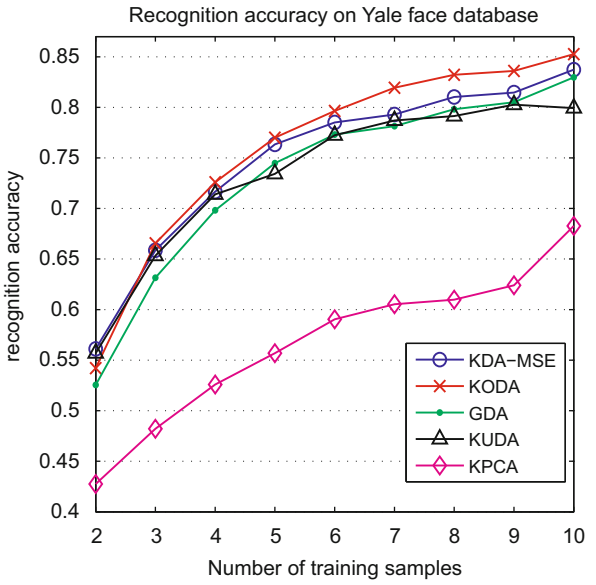

Fig. 7. Average recognition accuracies on Yale face database.

When the number of training samples is large, all of these kernel discriminant methods exhibit good performance. However, when the number of training samples is relatively small, the average accuracies of KDA-MSE and KODA are much higher than that of GDA, which implies that KDA-MSE and KODA are more effective for small sample size problem.

The average accuracy obtained by KPCA is the lowest in the classification experiments of ORL and Yale database. The reason is that the principal component vectors only aims to preserve the maximum energy in the reduced space rather than to keep or enhance the between-classes discriminatory power.

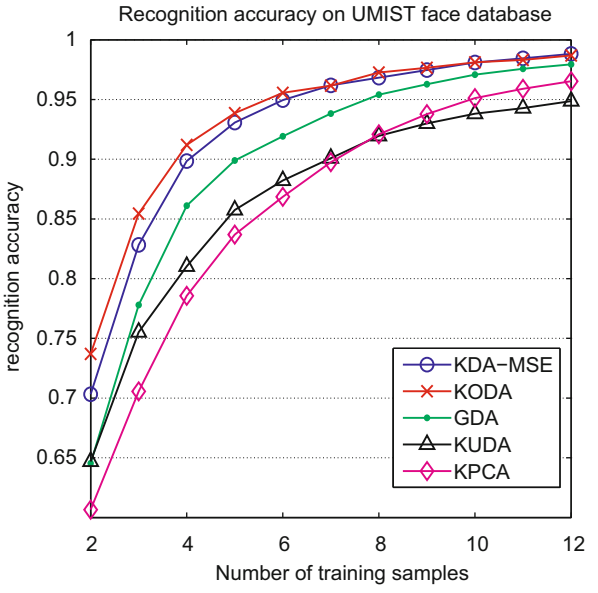

Fig. 8. Average recognition accuracies on UMIST face database.

Table 1
Description of two UCI datasets.

Dataset	Isolet	Mfeature
No. of class	26	10
Dimension	617	649
No. of sample	7797	2000

Table 2
Average recognition accuracies on UCI datasets.

	Isolet	Mfeature
KPCA	0.842	0.867
GDA	0.923	0.938
KUDA	0.942	0.964
KODA	0.963	0.981
KDA-MSE	0.966	0.975

4.3. *UCI datasets*

For the third experiment, we select two high-dimensional datasets from the UCI machine learning repository [34]. The detailed information of two selected datasets, i.e., Isolet and Mfeature is shown in Table 1.

Note that the number of samples of each class is smaller than the dimension; therefore it belongs to the undersampled problem for both of the two UCI datasets. The data are randomly splitted into equally sized training and test sets. We perform 50 independent runs to obtain the average recognition accuracy. Table 2 shows the recognition accuracies of the five kernel methods. It can be seen that the KDA-MSE algorithm has better performance than KPCA, GDA and KUDA, and is very competitive in comparison with KODA.

5. Conclusion

A new nonlinear dimensionality reduction algorithm, named KDA-MSE, is presented for enhancing discriminatory power. By exploiting the success of the kernel method in handling nonlinearly distributed data and the advantage of the MSE solution in dealing with undersampled problem, KDA-MSE overcomes two fundamental limitations of the conventional LDA. In addition to the applicability for small sample size data and nonlinearly separable data, there are three appealing properties of the proposed KDA-MSE. First it is suitable for multiclass situations. Second there is no additional regularization procedure and the regularization parameter adjustment can be avoided compared with other twoclass MSE-based kernel discriminant algorithms. Thirdly KDA-MSE saves about half calculation amount compared with GDA, KDDA and KUDA. Experimental results indicate that KDA-MSE achieves superior performance in terms of classification accuracy.

Despite these advantages, there still remain some problems to be solved. One problem is how to adjust the kernel parameters optimally, which is a hard and challenging task in kernel-based learning machines. Besides, since the size of kernel matrix is in proportion to the number of samples, the proposed KDA-MSE algorithm could be slow when the dataset is large. Both of the two problems not only exist in our algorithm, but also exist in the most kernel-based discriminant methods.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive advices that greatly improved the quality of this paper.

The COIL-20 dataset is from Columbia Object Image Library, Department of Computer Science Columbia University, New York, USA. The ORL, Yale and UMIST face datasets are from Olivetti Research Laboratory, Cambridge, UK, Center for Computation Vision and Control, Department of Computer Science Yale University, and Graham, respectively. The UCI dataset is from UC Irvine Machine Learning Repository.

Appendix A. Computation of $n(\mathbf{H}_b^{\Phi})^T\tilde{\Phi}(\mathbf{X})$

In this appendix, we derive Eqs. (64) and (65). First, let us introduce some submatrices for the convenience of derivation. The centered kernel matrix $\tilde{\mathbf{K}}$ can be partitioned into c submatrices by considering the class label

$$\tilde{\mathbf{K}}=\begin{bmatrix}\tilde{\mathbf{K}}^{(1)}\\ \tilde{\mathbf{K}}^{(2)}\\ \vdots\\ \tilde{\mathbf{K}}^{(c)}\end{bmatrix}, \tag{71}$$

where $n_i\times n$ submatrix $\tilde{\mathbf{K}}^{(i)}$ is the kernel matrix of the i-th class centered data and is defined by

$$\tilde{\mathbf{K}}^{(i)}=\tilde{\Phi}^T(\mathbf{X}^{(i)})\tilde{\Phi}(\mathbf{X}). \tag{72}$$

It is clear that the (k,j)-th entry of $\tilde{\mathbf{K}}^{(i)}$ is

$$[\tilde{\mathbf{K}}^{(i)}]_{kj}=\tilde{\Phi}^T(\mathbf{x}_k^{(i)})\tilde{\Phi}(\mathbf{x}_j),\quad 1\le k\le n_i,\ 1\le j\le n. \tag{73}$$

According to (43), the i-th row of matrix $(\mathbf{H}_b^{\Phi})^T$ is

$$\sqrt{\frac{n_i}{n}}(\mathbf{m}_i^{\tilde{\Phi}})^T=\sqrt{\frac{1}{nn_i}}\sum_{k=1}^{n_i}\tilde{\Phi}^T(\mathbf{x}_k^{(i)})$$

and according to (35), the j-th row of $\tilde{\Phi}(\mathbf{X})$ is $\tilde{\Phi}(\mathbf{x}_j)$. Since the (i,j)-th entry of matrix $n(\mathbf{H}_b^{\Phi})^T\tilde{\Phi}(\mathbf{X})$ is given by the product of the i-th row of $n(\mathbf{H}_b^{\Phi})^T$ and the j-th row of $\tilde{\Phi}(\mathbf{X})$, we have

$$\begin{aligned}[n(\mathbf{H}_b^{\Phi})^T\tilde{\Phi}(\mathbf{X})]_{ij}&=\sqrt{\frac{n}{n_i}}\sum_{k=1}^{n_i}\tilde{\Phi}^T(\mathbf{x}_k^{(i)})\tilde{\Phi}(\mathbf{x}_j)\\&=\sqrt{\frac{n}{n_i}}\sum_{k=1}^{n_i}[\tilde{\mathbf{K}}^{(i)}]_{kj}=\sqrt{\frac{n}{n_i}}[\mathbf{e}_i^T\tilde{\mathbf{K}}^{(i)}]_j,\end{aligned} \tag{74}$$

where $\mathbf{e}_i^T=[1,\ldots,1]$ is a row vector with size n_i. Eq. (74) means that

$$\begin{aligned}n(\mathbf{H}_b^{\Phi})^T\tilde{\Phi}(\mathbf{X})&=\begin{bmatrix}\sqrt{\frac{n}{n_1}}\mathbf{e}_1^T\tilde{\mathbf{K}}^{(1)}\\ \sqrt{\frac{n}{n_2}}\mathbf{e}_2^T\tilde{\mathbf{K}}^{(2)}\\ \vdots\\ \sqrt{\frac{n}{n_c}}\mathbf{e}_c^T\tilde{\mathbf{K}}^{(c)}\end{bmatrix}\\&=\sqrt{n}\begin{bmatrix}\frac{1}{\sqrt{n_1}}\mathbf{e}_1^T & & & \\ & \frac{1}{\sqrt{n_2}}\mathbf{e}_2^T & & \\ & & \ddots & \\ & & & \frac{1}{\sqrt{n_c}}\mathbf{e}_c^T\end{bmatrix}\begin{bmatrix}\tilde{\mathbf{K}}^{(1)}\\ \tilde{\mathbf{K}}^{(2)}\\ \vdots\\ \tilde{\mathbf{K}}^{(c)}\end{bmatrix}\\&=\mathbf{E}\tilde{\mathbf{K}},\end{aligned} \tag{75}$$

where the $c\times n$ matrix $\mathbf{E}$ is block diagonal.

Appendix B. Computation of $\tilde{\mathbf{k}}_z$

In this appendix, we prove the expression of $\tilde{\mathbf{k}}_z$ as shown in (67). Since $\tilde{\mathbf{k}}_z=\tilde{\Phi}^T(\mathbf{X})\tilde{\Phi}(\mathbf{z})$, the i-th entry of $\tilde{\mathbf{k}}_z$ equals the product of the i-th row of $\tilde{\Phi}^T(\mathbf{X})$ and $\tilde{\Phi}(\mathbf{z})$, i.e.,

$$\begin{aligned}[\tilde{\mathbf{k}}_z]_i&=\tilde{\Phi}^T(\mathbf{x}_i)\tilde{\Phi}(\mathbf{z})\\&=\left(\Phi^T(\mathbf{x}_i)-\frac{1}{n}\sum_{k=1}^{n}\Phi^T(\mathbf{x}_k)\right)\left(\Phi(\mathbf{z})-\frac{1}{n}\sum_{l=1}^{n}\Phi(\mathbf{x}_l)\right)\\&=\Phi^T(\mathbf{x}_i)\Phi(\mathbf{z})-\frac{1}{n}\sum_{l=1}^{n}\Phi^T(\mathbf{x}_i)\Phi(\mathbf{x}_l)-\frac{1}{n}\sum_{k=1}^{n}\Phi^T(\mathbf{x}_k)\Phi(\mathbf{z})\\&\quad+\frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n}\Phi^T(\mathbf{x}_k)\Phi(\mathbf{x}_l)\\&=\kappa(\mathbf{x}_i,\mathbf{z})-\frac{1}{n}\sum_{l=1}^{n}\kappa(\mathbf{x}_i,\mathbf{x}_l)-\frac{1}{n}\sum_{k=1}^{n}\kappa(\mathbf{x}_k,\mathbf{z})\\&\quad+\frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n}\kappa(\mathbf{x}_k,\mathbf{x}_l)\\&=[\mathbf{k}_z]_i-\sum_{l=1}^{n}[\mathbf{K}]_{il}\frac{1}{n}-\sum_{k=1}^{n}\frac{1}{n}[\mathbf{k}_z]_k+\sum_{k=1}^{n}\sum_{l=1}^{n}\frac{1}{n}[\mathbf{K}]_{kl}\frac{1}{n}.\end{aligned} \tag{76}$$

Define $n \times n$ matrix $\mathbf{1}_{n\times n}$ and $n \times 1$ vector $\mathbf{1}_{n\times 1}$ with all of their elements being $1/n$, then (76) can be written as

$$\begin{aligned}[\tilde{\mathbf{k}}_z]_i &= [\mathbf{k}_z]_i - \sum_{l=1}^{n}[\mathbf{K}]_{il}[\mathbf{1}_{n\times 1}]_l - \sum_{k=1}^{n}[\mathbf{1}_{n\times n}]_{ik}[\mathbf{k}_z]_k \\ &\quad + \sum_{k=1}^{n}\sum_{l=1}^{n}[\mathbf{1}_{n\times n}]_{ik}[\mathbf{K}]_{kl}[\mathbf{1}_{n\times 1}]_l \\ &= [\mathbf{k}_z]_i - [\mathbf{K}\mathbf{1}_{n\times 1}]_i - [\mathbf{1}_{n\times n}\mathbf{k}_z]_i + [\mathbf{1}_{n\times n}\mathbf{K}\mathbf{1}_{n\times 1}]_i \\ &= [\mathbf{k}_z - \mathbf{K}\mathbf{1}_{n\times 1} - \mathbf{1}_{n\times n}\mathbf{k}_z + \mathbf{1}_{n\times n}\mathbf{K}\mathbf{1}_{n\times 1}]_i \end{aligned} \tag{77}$$

which means that $\tilde{\mathbf{k}}_z = \mathbf{k}_z - \mathbf{K}\mathbf{1}_{n\times 1} - \mathbf{1}_{n\times n}\mathbf{k}_z + \mathbf{1}_{n\times n}\mathbf{K}\mathbf{1}_{n\times 1}$ holds true.

References

[1] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic, New York, 1990.

[2] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley-Interscience, New York, 2001.

[3] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (July 1997) 711–720.

[4] J.H. Friedman, Regularized discriminant analysis, Journal of the American Statistical Association 84 (405) (1989) 165–175.

[5] T. Hastie, A. Buja, R. Tibshirani, Penalized discriminant analysis, Annals of Statistics 23 (1995) 3–102.

[6] L.-F. Chen, H.-Y.M. Liao, M.-T. Ko, J.-C. Lin, A new LDA-based face recognition system which can solve the small sample size problem, Pattern Recognition 33 (10) (October 2000) 1713–1726.

[7] H. Yu, J. Yang, A direct LDA algorithm for high-dimensional data—with application to face recognition, Pattern Recognition 34 (10) (October 2001) 2067–2070.

[8] P. Howland, M. Jeon, H. Park, Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition, SIAM Journal on Matrix Analysis and Applications 25 (1) (2003) 165–179.

[9] H. Li, T. Jiang, K. Zhang, Efficient and robust feature extraction by maximum margin criterion, IEEE Transactions on Neural Networks 17 (1) (January 2006) 157–165.

[10] Z. Jin, J.Y. Yang, Z.S. Hu, Z. Lou, Face recognition based on the uncorrelated discriminant transformation, Pattern Recognition 34 (7) (July 2001) 1405–1416.

[11] J. Ye, Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems, Journal of Machine Learning Research 6 (April 2005) 483–502.

[12] C. Park, H. Park, A relationship between linear discriminant analysis and the generalized minimum squared error solution, SIAM Journal on Matrix Analysis and Applications 27 (2) (2005) 474–492.

[13] J. Ye, Least squares linear discriminant analysis, in: Proceedings of International Conference on Machine Learning, 2007, pp. 1087–1093.

[14] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[15] K.R. Müler, S. Mika, G. Räsch, K. Tsuda, B. Schökopf, An introduction to kernel-based learning algorithms, IEEE Transactions on Neural Networks 12 (3) (March 2001) 181–201.

[16] B. Schökopf, A. Smola, K.R. Müler, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (5) (July 1998) 1299–1319.

[17] S. Mika, G. Räsch, J. Weston, B. Schökopf, K.R. Müler, Fisher discriminant analysis with kernels, in: Proceedings of IEEE International Workshop on Neural Networks for Signal Processing, 1999, pp. 41–48.

[18] S. Mika, Kernel Fisher discriminants, Ph.D. Dissertation, University of Technology, Berlin, Germany, 2002.

[19] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Computation 12 (10) (October 2000) 2385–2404.

[20] C. Park, H. Park, Nonlinear discriminant analysis using kernel functions and the generalized singular value decomposition, SIAM Journal on Matrix Analysis and Applications 27 (1) (2005) 87–102.

[21] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, Face recognition using kernel direct discriminant analysis algorithms, IEEE Transactions on Neural Networks 14 (1) (January 2003) 117–126.

[22] J. Yang, A.F. Frangi, J. Yang, D. Zhang, Z. Jin, KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (2) (February 2005) 230–244.

[23] Q. Liu, X. Tang, H. Lu, S. Ma, Face recognition using kernel scatter-difference-based discriminant analysis, IEEE Transactions on Neural Networks 17 (4) (July 2006) 1081–1085.

[24] T. Xiong, J. Ye, Kernel uncorrelated and orthogonal discriminant analysis: a unified approach, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 125–131.

[25] G.H. Golub, C.F. Van loan, Matrix Computations, Johns Hopkins University Press, Baltimore, MD, 1996.

[26] S. Billings, K. Lee, Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm, Neural Networks 15 (2) (March 2002) 263–270.

[27] A. Ruiz, P.E. López-de-Teruel, Nonlinear kernel-based statistical pattern analysis, IEEE Transactions on Neural Networks 12 (1) (January 2001) 16–32.

[28] H. Kim, B.L. Drake, H. Park, Adaptive nonlinear discriminant analysis by regularized minimum squared errors, IEEE Transactions on Knowledge and Data Engineering 18 (5) (May 2006) 603–612.

[29] R. Fisher, The use of multiple measurements in taxonomic problems, Annals of Eugenics 7 (1936) 179–188.

[30] S.A. Nene, S.K. Nayar, H. Murase, Columbia object image library (COIL-20), Columbia University, Technical Report CUCS-005-96, 1996.

[31] F. Samaria, A. Harter, Parameterisation of a stochastic model for human face identification, in: Proceedings of Second IEEE Workshop on Applications of Computer Vision, 1994.

[32] Yale University Face Database ⟨http://cvc.yale.edu/projects/yalefaces/yalefaces.html⟩, 2002.

[33] D.B. Graham, N.M. Allinson, Characterizing virtual eigensignatures for general purpose face recognition, in: H. Wechsler, P.J. Phillips, V. Bruce, F. Fogelman-Soulie, T.S. Huang (Eds.), Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences, vol. 163, 1998, pp. 446–456.

[34] UCI Machine Learning Repository ⟨http://archive.ics.uci.edu/ml/⟩.