Kernel Discriminant Learning with Application to Face Recognition

Juwei Lu[1], K.N. Plataniotis[2], and A.N. Venetsanopoulos[3]

Bell Canada Multimedia Laboratory
The Edward S. Rogers Sr. Department of Electrical and Computer Engineering
University of Toronto, Toronto, M5S 3G4, Ontario, Canada
Emails: {juwei[1], kostas[2], anv[3]}@dsp.toronto.edu

Abstract. When applied to high-dimensional pattern classification tasks such as face recognition, traditional kernel discriminant analysis methods often suffer from two problems: 1) small training sample size compared to the dimensionality of the sample (or mapped kernel feature) space, and 2) high computational complexity. In this chapter, we introduce a new kernel discriminant learning method, which attempts to deal with the two problems by using regularization and subspace decomposition techniques. The proposed method is tested by extensive experiments performed on real face databases. The obtained results indicate that the method outperforms, in terms of classification accuracy, existing kernel methods, such as kernel Principal Component Analysis and kernel Linear Discriminant Analysis, at a significantly reduced computational cost.

Keywords: Statistical Discriminant Analysis, Kernel Machines, Small Sample Size, Nonlinear Feature Extraction, Face Recognition

1 Introduction

Statistical learning theory tells us essentially that the difficulty of an estimation problem increases drastically with the dimensionality J of the sample space, since in principle, as a function of J, one needs exponentially many patterns to sample the space properly [18, 32]. Unfortunately, in many practical tasks such as face recognition, the number of available training samples per subject is usually much smaller than the dimensionality of the sample space. For instance, a canonical example used for face recognition is a 112×92 image, which exists in a 10304-dimensional real space. Nevertheless, the number of examples per class available for learning is not more than ten in most cases. This results in the so-called *small sample size* (SSS) problem, which is known to have significant influences on the performance of a statistical pattern recognition system (see *e.g.* [3, 5, 9, 12, 13, 16, 21, 33, 34]).

When it comes to statistical discriminant learning tasks such as Linear Discriminant Analysis (LDA), the SSS problem often gives rise to high variance in the estimation for the between- and within-class scatter matrices, which are either poorly- or ill-posed. To address the problem, one popular approach is to introduce an intermediate Principal Component Analysis (PCA) step to remove the null spaces of the two scatter matrices. LDA is then performed in the lower dimensional PCA subspace, as it was done for example in [3, 29]. However, it has been shown that the discarded null spaces may contain significant discriminatory information [10]. To prevent this from happening, solutions without a separate PCA step, called *direct* LDA (D-LDA) approaches have been presented recently in [5, 12, 34]. The underlying principle behind the approaches is that the information residing in (or close to) the null space of the within-class scatter matrix is more significant for discriminant tasks than the information out of (or far away from) the null space. Generally, the null space of a matrix is determined by its zero eigenvalues. However, due to insufficient training samples, it is very difficult to identify the true null eigenvalues. As a result, high variance is often introduced in the estimation for the zero (or very small) eigenvalues of the within-class scatter matrix. Note that the eigenvectors corresponding to these eigenvalues are considered the most significant feature bases in the D-LDA approaches [5, 12, 34].

In this chapter, we study statistical discriminant learning algorithms in some high-dimensional feature space, mapped from the input sample space by the so-called "kernel machine" technique [18, 22, 25, 32]. In the feature space, it is hoped that the distribution of the mapped data is simplified, so that traditional linear methods can perform well. A problem with the idea is that the dimensionality of the feature space may be extremely higher than that of the sample space, resulting in the introduction of the SSS problem, or the worse if it has existed. In addition, kernel-based algorithms are generally much more computationally expensive compared to their linear counterparts. To address these problems, we introduce a regularized discriminant analysis method in the kernel feature space. This method deals with the SSS problem under the D-LDA framework of [12, 34]. Nevertheless, it is based on a modified Fisher's discriminant criterion specifically designed to avoid the unstable problem with the approach of [34]. Also, a side-effect of the design is that the computational complexity is significantly reduced compared to other two popular kernel methods, kernel PCA (KPCA) [26] and kernel LDA (GDA) [2]. The effectiveness of the presented method is demonstrated in the face recognition application.

2 Kernel-based Statistical Pattern Analysis

In the statistical pattern recognition tasks, the problem of feature extraction can be stated as follows: Assume that we have a training set, $\mathcal{Z} = \{\mathcal{Z}_i\}_{i=1}^{C}$, containing C classes with each class $\mathcal{Z}_i = \{\mathbf{z}_{ij}\}_{j=1}^{C_i}$ consisting of a number of

examples $\mathbf{z}_{ij} \in \mathbb{R}^J$, where $\mathbb{R}^J$ denotes the J-dimensional real space. Taking as input such a set $\mathcal{Z}$, the objective of learning is to find, based on optimization of certain separability criteria, a transformation φ which produces a feature representation $\mathbf{y}_{ij} = \varphi(\mathbf{z}_{ij})$, $\mathbf{y}_{ij} \in \mathbb{R}^M$, intrinsic to the objects of these examples with enhanced discriminatory power.

2.1 Input Sample Space *vs* Kernel Feature Space

The kernel machines provide an elegant way of designing nonlinear algorithms by reducing them to linear ones in some high-dimensional feature space $\mathbb{F}$ nonlinearly related to the input sample space $\mathbb{R}^J$:

$$\phi : \mathbf{z} \in \mathbb{R}^J \rightarrow \phi(\mathbf{z}) \in \mathbb{F} \tag{1}$$

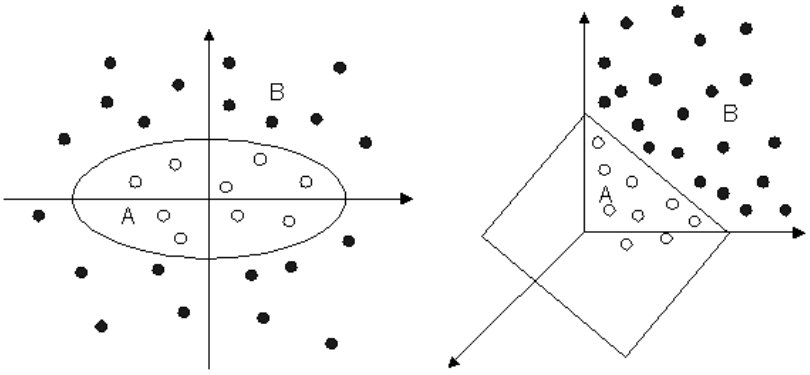

Fig. 1. A toy example of two-class pattern classification problem [27]. **Left:** samples lie in the 2-D input space, where it needs a nonlinear ellipsoidal decision boundary to separate classes A and B. **Right:** Samples are mapped to a 3-D feature space, where a linear hyperplane can separate the two classes.

The idea can be illustrated by a toy example depicted in Fig.1, where two-dimensional input samples, say $\mathbf{z} = [z_1, z_2]$, are mapped to a three-dimensional feature space through a nonlinear transform: $\phi : \mathbf{z} = [z_1, z_2] \rightarrow \phi(\mathbf{z}) = [x_1, x_2, x_3] := \left[z_1^2, \sqrt{2}z_1 z_2, z_2^2\right]$ [27]. It can be seen from Fig.1 that in the sample space, a nonlinear ellipsoidal decision boundary is needed to separate classes A and B, in contrast with this, the two classes become linearly separable in the higher-dimensional feature space.

The feature space $\mathbb{F}$ could be regarded as a "linearization space" [1]. However, to reach this goal, its dimensionality could be arbitrarily large, possibly infinite. Fortunately, the exact $\phi(\mathbf{z})$ is not needed and the feature space can become implicit by using kernel machines. The trick behind the methods is to

replace dot products in $\mathbb{F}$ with a kernel function in the input space $\mathbb{R}^J$ so that the nonlinear mapping is performed implicitly in $\mathbb{R}^J$. Let us come back to the toy example of Fig.1, where the feature space is spanned by the second-order monomials of the input sample. Let $\mathbf{z}_i \in \mathbb{R}^2$ and $\mathbf{z}_j \in \mathbb{R}^2$ be two examples in the input space, and the dot product of their feature vectors $\phi(\mathbf{z}_i) \in \mathbb{F}$ and $\phi(\mathbf{z}_j) \in \mathbb{F}$ can be computed by the following kernel function, $k(\mathbf{z}_i, \mathbf{z}_j)$, defined in $\mathbb{R}^2$,

$$\begin{aligned} \phi(\mathbf{z}_i) \cdot \phi(\mathbf{z}_j) &= \left[z_{i1}^2, \sqrt{2} z_{i1} z_{i2}, z_{i2}^2\right] \left[z_{j1}^2, \sqrt{2} z_{j1} z_{j2}, z_{j2}^2\right]^T \\ &= \left([z_{i1}, z_{i2}] \, [z_{j1}, z_{j2}]^T\right)^2 = (\mathbf{z}_i \cdot \mathbf{z}_j)^2 =: k(\mathbf{z}_i, \mathbf{z}_j) \end{aligned} \tag{2}$$

From this example, it can be seen that the central issue to generalize a linear learning algorithm to its kernel version is to reformulate all the computations of the algorithm in the feature space in the form of dot product. Based on the properties of the kernel functions used, the kernel generation gives rise to neural-network structures, splines, Gaussian, Polynomial or Fourier expansions, *etc.* . Any function satisfying Mercer's condition [17] can be used as a kernel. Table 1 lists some of the most widely used kernel functions, and more sophisticated kernels can be found in [24, 27, 28, 36].

Table 1. Some of the most widely used kernel functions, where $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^J$.

Gaussian RBF	$k(\mathbf{z}_1, \mathbf{z}_2) = \exp\left(\frac{-\|\mathbf{z}_1 - \mathbf{z}_2\|^2}{\sigma^2}\right)$, $\sigma \in \mathbb{R}$
Polynomial	$k(\mathbf{z}_1, \mathbf{z}_2) = (a(\mathbf{z}_1 \cdot \mathbf{z}_2) + b)^d$, $a \in \mathbb{R}$, $b \in \mathbb{R}$, $d \in \mathbb{N}$
Sigmoidal	$k(\mathbf{z}_1, \mathbf{z}_2) = \tanh(a(\mathbf{z}_1 \cdot \mathbf{z}_2) + b)$, $a \in \mathbb{R}$, $b \in \mathbb{R}$
Inverse multiquadric	$1/\sqrt{\|\mathbf{z}_1 - \mathbf{z}_2\|^2 + \sigma^2}$, $\sigma \in \mathbb{R}$

2.2 Kernel Principal Component Analysis (KPCA)

To find principal components of a non convex distribution, the classic PCA has been generalized to the *kernel* PCA (KPCA) [26]. Given the nonlinear mapping of Eq.1, the covariance matrix of the training sample $\mathcal{Z}$ in the feature space $\mathbb{F}$ can be expressed as

$$\tilde{\mathbf{S}}_{cov} = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{C_i} (\phi(\mathbf{z}_{ij}) - \bar{\phi})(\phi(\mathbf{z}_{ij}) - \bar{\phi})^T \tag{3}$$

where $N = \sum_{i=1}^{C} C_i$, and $\bar{\phi} = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{C_i} \phi(\mathbf{z}_{ij})$ is the average of the ensemble in $\mathbb{F}$. The KPCA is actually a classic PCA performed in the feature space $\mathbb{F}$. Let $\tilde{\mathbf{g}}_m \in \mathbb{F}$ $(m = 1, 2, \cdots, M)$ be the first M most significant eigenvectors of $\tilde{\mathbf{S}}_{cov}$, and they form a low-dimensional subspace, called "KPCA subspace" in $\mathbb{F}$. All these $\{\tilde{\mathbf{g}}_m\}_{m=1}^{M}$ lie in the span of $\{\phi(\mathbf{z}_{ij})\}_{\mathbf{z}_{ij} \in \mathcal{Z}}$, and have

$\tilde{\mathbf{g}}_m = \sum_{i=1}^{C}\sum_{j=1}^{C_i} a_{ij}\phi(\mathbf{z}_{ij})$, where a_{ij} are the linear combination coefficients. For any input pattern $\mathbf{z}$, its nonlinear principal components can be obtained by the dot product, $\mathbf{y}_m = \tilde{\mathbf{g}}_m \cdot (\phi(\mathbf{z}) - \bar{\phi})$, computed indirectly through a kernel function $k()$.

2.3 Generalized Discriminant Analysis (GDA)

As such, *Generalized Discriminant Analysis* (GDA, also known as kernel LDA) [2] is a process to extract a nonlinear discriminant feature representation by performing a classic LDA in the high-dimensional feature space $\mathbb{F}$. Let $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$ be the between- and within-class scatter matrices in the feature space $\mathbb{F}$ respectively, and they have following expressions:

$$\tilde{\mathbf{S}}_b = \frac{1}{N}\sum_{i=1}^{C} C_i(\bar{\phi}_i - \bar{\phi})(\bar{\phi}_i - \bar{\phi})^T \tag{4}$$

$$\tilde{\mathbf{S}}_w = \frac{1}{N}\sum_{i=1}^{C}\sum_{j=1}^{C_i} (\phi(\mathbf{z}_{ij}) - \bar{\phi}_i)(\phi(\mathbf{z}_{ij}) - \bar{\phi}_i)^T \tag{5}$$

where $\bar{\phi}_i = \frac{1}{C_i}\sum_{j=1}^{C_i}\phi(\mathbf{z}_{ij})$ is the mean of class $\mathcal{Z}_i$. In the same way as LDA, GDA determines a set of optimal nonlinear discriminant basis vectors by maximizing the standard Fisher's criterion:

$$\tilde{\Psi} = \arg\max_{\tilde{\Psi}} \frac{\left|\tilde{\Psi}^T\tilde{\mathbf{S}}_b\tilde{\Psi}\right|}{\left|\tilde{\Psi}^T\tilde{\mathbf{S}}_w\tilde{\Psi}\right|}, \quad \tilde{\Psi} = [\tilde{\psi}_1, \cdots, \tilde{\psi}_M], \quad \tilde{\psi}_m \in \mathbb{F} \tag{6}$$

Similar to KPCA, the GDA-based feature representation of an input pattern $\mathbf{z}$ can be obtained by a linear projection in $\mathbb{F}$, $\mathbf{y}_m = \tilde{\psi}_m \cdot \mathbf{z}$.

From the above presentation, it can be seen that KPCA and GDA are based on the exactly same optimization criteria to their linear counterparts, PCA and LDA. Especially, KPCA and GDA reduce to PCA and LDA, respectively, when $\phi(\mathbf{z}) = \mathbf{z}$. As we know, LDA optimizes the low-dimensional representation of the objects with focus on the most discriminant feature extraction while PCA achieves simply object reconstruction in a least-square sense. The difference may lead to significantly different orientations of feature bases as shown in Fig.2:**Left**, where it is not difficult to see that the representation obtained by PCA is entirely unsuitable for the task of separating the two classes. As a result, it is generally believed that when it comes to solving problems of pattern classification, the LDA-based feature representation is usually superior to the PCA-based one [3, 5, 34].

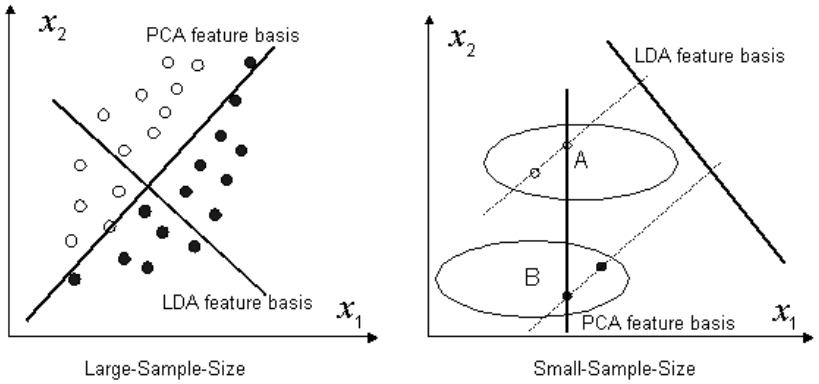

Fig. 2. PCA vs LDA in different learning scenarios. **Left:** given a large size sample of two classes, LDA finds a much better feature basis than PCA for the classification task. **Right:** given a small size sample of two classes, LDA gets over-fitting, and is outperformed by PCA [15].

3 Discriminant Learning in Small-Sample-Size Scenarios

For simplicity, we start the discussion with the linear case of discriminant learning, *i.e.* LDA, which optimizes the criterion of Eq.6 in the sample space $\mathbb{R}^J$. This is equivalent to setting $\phi(\mathbf{z}) = \mathbf{z}$ during the GDA process.

3.1 The Small-Sample-Size (SSS) problem

As mentioned in Section 1, the so-called Small-Sample-Size (SSS) problem is often introduced when LDA is carried out in some high-dimensional space. Compared to the PCA solution, the LDA solution is much more susceptible to the SSS problem given the same training set, since the latter requires many more training samples than the former due to the increased number of parameters needed to be estimated [33]. Especially when the number of available training samples is less than the dimensionality of the space, the two scatter matrix estimates, $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$, are highly ill-posed and singular. As a result, the general belief that LDA is superior to PCA in the context of pattern classification may not be correct in the SSS scenarios [15]. The phenomenon of LDA over-fitting the training data in the SSS settings can be illustrated by a simple example shown in Fig.2:**Right**, where PCA yields a superior feature basis for the purpose of pattern classification [15].

3.2 Where are the optimal discriminant features?

When $\tilde{\mathbf{S}}_w$ is non-singular, the basis vectors $\tilde{\Psi}$ sought in Eq.6 correspond to the first M most significant eigenvectors of $(\tilde{\mathbf{S}}_w^{-1}\tilde{\mathbf{S}}_b)$, where the "significant"

means that the eigenvalues corresponding to these eigenvectors are the first M largest ones. However, due to the SSS problem, often an extremely singular $\tilde{\mathbf{S}}_w$ is generated when $N \ll J$. Let us assume that $\mathcal{A}$ and $\mathcal{B}$ represent the null spaces of $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$ respectively, while $\mathcal{A}' = \mathbb{R}^J - \mathcal{A}$ and $\mathcal{B}' = \mathbb{R}^J - \mathcal{B}$ denote the orthogonal complements of $\mathcal{A}$ and $\mathcal{B}$. Traditional approaches attempt to solve the problem by utilizing an intermediate PCA step to remove $\mathcal{A}$ and $\mathcal{B}$. LDA is then performed in the lower dimensional PCA subspace, as it was done for example in [3,29]. Nevertheless, it should be noted at this point that the maximum of the ratio in Eq.6 can be reached only when $\left|\tilde{\Psi}^T \tilde{\mathbf{S}}_w \tilde{\Psi}\right| = 0$ and $\left|\tilde{\Psi}^T \tilde{\mathbf{S}}_b \tilde{\Psi}\right| \neq 0$. This means that the discarded null space $\mathcal{B}$ may contain the most significant discriminatory information. On the other hand, there is no significant information, in terms of the maximization in Eq.6, to be lost if $\mathcal{A}$ is discarded. It is not difficult to see at this point that when $\tilde{\Psi} \in \mathcal{A}$, the ratio $\frac{\left|\tilde{\Psi}^T \tilde{\mathbf{S}}_b \tilde{\Psi}\right|}{\left|\tilde{\Psi}^T \tilde{\mathbf{S}}_w \tilde{\Psi}\right|}$ drops to its minimum value, 0. Therefore, many researchers consider the intersection space $(\mathcal{A}' \cap \mathcal{B})$ to be spanned by the optimal discriminant feature bases [5,10].

Based on the above ideas, Yu and Yang proposed the so-called direct LDA (YD-LDA) approach in order to prevent the removal of useful discriminant information contained in the null space $\mathcal{B}$ [34]. However, it has been recently found that the YD-LDA performance may deteriorate rapidly due to two problems that may be encountered when the SSS problem becomes severe [13]. One problem is that the zero eigenvalues of the within-class scatter matrix are used as possible divisors, so that the YD-LDA process can not be carried out. The other is that the worse of the SSS situations may significantly increase the variance in the estimation for the small eigenvalues of the within-class scatter matrix, while the importance of the eigenvectors corresponding to these small eigenvalues is dramatically exaggerated.

The discussions given in these two Sections are based on LDA carried out in the sample space $\mathbb{R}^J$. When LDA comes to the feature space $\mathbb{F}$, it is not difficult to see that the SSS problem becomes worse essentially due to the much higher dimensionality. However, GDA, following traditional approach, attempts to solve the problem simply by removing the two null spaces, $\mathcal{A}$ and $\mathcal{B}$. As a result, it can be known from the above analysis that some significant discriminant information may be lost inevitably due to such a process.

4 Regularized Kernel Discriminant Learning (R-KDA)

To address the problems with the GDA and YD-LDA methods in the SSS scenarios, a *regularized kernel discriminant analysis* method, named R-KDA, is developed here.

4.1 A regularized Fisher's criterion

To this end, we first introduce a regularized Fisher's criterion [14]. The criterion, which is utilized in this work instead of the conventional one (Eq.6), can be expressed as follows:

$$\tilde{\Psi} = \arg\max_{\tilde{\Psi}} \frac{\left|\tilde{\Psi}^T \tilde{\mathbf{S}}_b \tilde{\Psi}\right|}{\left|\eta(\tilde{\Psi}^T \tilde{\mathbf{S}}_b \tilde{\Psi}) + (\tilde{\Psi}^T \tilde{\mathbf{S}}_w \tilde{\Psi})\right|} \tag{7}$$

where $0 \leq \eta \leq 1$ is a regularization parameter. Although Eq.7 looks different from Eq.6, it can be shown that the modified criterion is exactly equivalent to the conventional one by the following theorem.

Theorem 1. *Let $\mathbb{R}^J$ denote the J-dimensional real space, and suppose that $\forall \psi \in \mathbb{R}^J$, $u(\psi) \geq 0$, $v(\psi) \geq 0$, $u(\psi) + v(\psi) > 0$ and $0 \leq \eta \leq 1$. Let $q_1(\psi) = \frac{u(\psi)}{v(\psi)}$ and $q_2(\psi) = \frac{u(\psi)}{\eta \cdot u(\psi) + v(\psi)}$. Then, $q_1(\psi)$ has the maximum (including positive infinity) at point $\psi^* \in \mathbb{R}^J$ iff $q_2(\psi)$ has the maximum at point ψ^*.*

Proof. Since $u(\psi) \geq 0$, $v(\psi) \geq 0$ and $0 \leq \eta \leq 1$, we have $0 \leq q_1(\psi) \leq +\infty$ and $0 \leq q_2(\psi) \leq \frac{1}{\eta}$.

1. If $\eta = 0$, then $q_1(\psi) = q_2(\psi)$.
2. If $0 < \eta \leq 1$ and $v(\psi) = 0$, then $q_1(\psi) = +\infty$ and $q_2(\psi) = \frac{1}{\eta}$.
3. If $0 < \eta \leq 1$ and $v(\psi) > 0$, then

$$q_2(\psi) = \frac{\frac{u(\psi)}{v(\psi)}}{1 + \eta \frac{u(\psi)}{v(\psi)}} = \frac{q_1(\psi)}{1 + \eta q_1(\psi)} = \frac{1}{\eta}\left(1 - \frac{1}{1 + \eta q_1(\psi)}\right) \tag{8}$$

It can be seen from Eq.8 that $q_2(\psi)$ increases iff $q_1(\psi)$ increases.

Combining the above three cases, the theorem is proven.

The regularized Fisher's criterion is a function of the parameter η, which controls the strength of regularization. Within the variation range of η, two extremes should be noted. In one extreme where $\eta = 0$, the modified Fisher's criterion is reduced to the conventional one with no regularization. In contrast with this, strong regularization is introduced in another extreme where $\eta = 1$. In this case, Eq.7 becomes $\tilde{\Psi} = \arg\max_{\tilde{\Psi}} \frac{\left|\tilde{\Psi}^T \tilde{\mathbf{S}}_b \tilde{\Psi}\right|}{\left|(\tilde{\Psi}^T \tilde{\mathbf{S}}_b \tilde{\Psi}) + (\tilde{\Psi}^T \tilde{\mathbf{S}}_w \tilde{\Psi})\right|}$, which as a variant of the original Fisher's criterion has been also widely used for example in [5,10–12]. Among these examples, the method of [12] is a D-LDA variant with $\eta = 1$ (hereafter JD-LDA). The advantages of introducing the regularization will be seen during the development of the R-KDA method proposed in the following sections.

4.2 Eigen-analysis of $\tilde{\mathbf{S}}_b$ in the Feature Space $\mathbb{F}$

Following the D-LDA framework of [11,12], we start by solving the eigenvalue problem of $\tilde{\mathbf{S}}_b$, which can be rewritten here as follows,

$$\tilde{\mathbf{S}}_b = \sum_{i=1}^{C} \left(\sqrt{\frac{C_i}{N}} \left(\bar{\phi}_i - \bar{\phi} \right) \right) \left(\sqrt{\frac{C_i}{N}} \left(\bar{\phi}_i - \bar{\phi} \right) \right)^T = \sum_{i=1}^{C} \tilde{\bar{\phi}}_i \tilde{\bar{\phi}}_i^{\,T} = \tilde{\Phi}_b \tilde{\Phi}_b^T \qquad (9)$$

where $\tilde{\bar{\phi}}_i = \sqrt{\frac{C_i}{N}} \left(\bar{\phi}_i - \bar{\phi} \right)$ and $\tilde{\Phi}_b = \left[\tilde{\bar{\phi}}_1, \cdots, \tilde{\bar{\phi}}_c \right]$. Since the dimensionality of the feature space $\mathbb{F}$, denoted as J', could be arbitrarily large or possibly infinite, it is intractable to directly compute the eigenvectors of the $(J' \times J')$ matrix $\tilde{\mathbf{S}}_b$. Fortunately, the first m ($\leq C-1$) most significant eigenvectors of $\tilde{\mathbf{S}}_b$, corresponding to non-zero eigenvalues, can be indirectly derived from the eigenvectors of the matrix $\tilde{\Phi}_b^T \tilde{\Phi}_b$ (with size $C \times C$) [11].

To this end, we assume that there exists a kernel function $k(\mathbf{z}_i, \mathbf{z}_j) = \phi(\mathbf{z}_i) \cdot \phi(\mathbf{z}_j)$ for any $\phi(\mathbf{z}_i), \phi(\mathbf{z}_j) \in \mathbb{F}$, and then define an $N \times N$ dot product matrix $\mathbf{K}$,

$$\mathbf{K} = (K_{lh})_{\substack{l=1,\cdots,C \\ h=1,\cdots,C}} \text{ with } K_{lh} = (k_{ij})_{\substack{i=1,\cdots,C_l \\ j=1,\cdots,C_h}} \qquad (10)$$

where $k_{ij} = k(\mathbf{z}_{li}, \mathbf{z}_{hj}) = \phi_{li} \cdot \phi_{hj}$, $\phi_{li} = \phi(\mathbf{z}_{li})$ and $\phi_{hj} = \phi(\mathbf{z}_{hj})$. The matrix $\mathbf{K}$ allows us to express $\tilde{\Phi}_b^T \tilde{\Phi}_b$ as follows [11]:

$$\begin{aligned} \tilde{\Phi}_b^T \tilde{\Phi}_b = \; & \tfrac{1}{N} \mathbf{B} \cdot (\mathbf{A}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{A}_{NC} - \tfrac{1}{N} (\mathbf{A}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{1}_{NC}) - \\ & \tfrac{1}{N} (\mathbf{1}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{A}_{NC}) + \tfrac{1}{N^2} (\mathbf{1}_{NC}^T \cdot \mathbf{K} \cdot \mathbf{1}_{NC})) \cdot \mathbf{B} \end{aligned} \qquad (11)$$

where $\mathbf{B} = \mathbf{diag}\left[\sqrt{C_1}, \cdots, \sqrt{C_c} \right]$, $\mathbf{1}_{NC}$ is an $N \times C$ matrix with terms all equal to one, $\mathbf{A}_{NC} = \mathbf{diag}\left[\mathbf{a}_{c_1}, \cdots, \mathbf{a}_{c_c} \right]$ is an $N \times C$ block diagonal matrix, and $\mathbf{a}_{c_i}$ is a $C_i \times 1$ vector with all terms equal to: $\frac{1}{C_i}$.

Let $\tilde{\lambda}_i$ and $\tilde{\mathbf{e}}_i$ ($i = 1, \cdots, C$) be the i-th eigenvalue and its corresponding eigenvector of $\tilde{\Phi}_b^T \tilde{\Phi}_b$, sorted in **decreasing** order of the eigenvalues. Since $(\tilde{\Phi}_b \tilde{\Phi}_b^T)(\tilde{\Phi}_b \tilde{\mathbf{e}}_i) = \tilde{\lambda}_i (\tilde{\Phi}_b \tilde{\mathbf{e}}_i)$, $\tilde{\mathbf{v}}_i = \tilde{\Phi}_b \tilde{\mathbf{e}}_i$ is an eigenvector of $\tilde{\mathbf{S}}_b$. In order to remove the null space of $\tilde{\mathbf{S}}_b$, we only use its first m ($\leq C-1$) eigenvectors: $\tilde{\mathbf{V}} = [\tilde{\mathbf{v}}_1, \cdots, \tilde{\mathbf{v}}_m] = \tilde{\Phi}_b \tilde{\mathbf{E}}_m$ with $\tilde{\mathbf{E}}_m = [\tilde{\mathbf{e}}_1, \cdots, \tilde{\mathbf{e}}_m]$, whose corresponding eigenvalues are greater than 0. It is not difficult to see that $\tilde{\mathbf{V}}^T \tilde{\mathbf{S}}_b \tilde{\mathbf{V}} = \tilde{\Lambda}_b$, with $\tilde{\Lambda}_b = \mathbf{diag}[\tilde{\lambda}_1^2, \cdots, \tilde{\lambda}_m^2]$, an $(m \times m)$ diagonal matrix.

4.3 Eigen-analysis of $\tilde{\mathbf{S}}_w$ in the Feature Space $\mathbb{F}$

Let $\tilde{\mathbf{U}} = \tilde{\mathbf{V}} \tilde{\Lambda}_b^{-1/2}$, each column vector of which lies in the feature space $\mathbb{F}$. Projecting both $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$ into the subspace spanned by $\tilde{\mathbf{U}}$, it can be easily seen that $\tilde{\mathbf{U}}^T \tilde{\mathbf{S}}_b \tilde{\mathbf{U}} = \mathbf{I}$, an $(m \times m)$ identity matrix, while $\tilde{\mathbf{U}}^T \tilde{\mathbf{S}}_w \tilde{\mathbf{U}}$ can be expanded as:

$$\tilde{\mathbf{U}}^T \tilde{\mathbf{S}}_w \tilde{\mathbf{U}} = (\tilde{\mathbf{E}}_m \tilde{\Lambda}_b^{-1/2})^T (\tilde{\Phi}_b^T \tilde{\mathbf{S}}_w \tilde{\Phi}_b) (\tilde{\mathbf{E}}_m \tilde{\Lambda}_b^{-1/2}) \qquad (12)$$

Using the kernel matrix $\mathbf{K}$, a closed form expression of $\tilde{\Phi}_b^T \tilde{\mathbf{S}}_w \tilde{\Phi}_b$ can be obtained as follows [11],

$$\begin{aligned}\tilde{\Phi}_b^T \tilde{\mathbf{S}}_w \tilde{\Phi}_b = \quad & \tfrac{1}{N^2}\mathbf{B}\cdot(\mathbf{A}_{NC}^T\cdot\hat{\mathbf{K}}\cdot\mathbf{A}_{NC} - \tfrac{1}{N}(\mathbf{A}_{NC}^T\cdot\hat{\mathbf{K}}\cdot\mathbf{1}_{NC}) - \\ & \tfrac{1}{N}(\mathbf{1}_{NC}^T\cdot\hat{\mathbf{K}}\cdot\mathbf{A}_{NC}) + \tfrac{1}{N^2}(\mathbf{1}_{NC}^T\cdot\hat{\mathbf{K}}\cdot\mathbf{1}_{NC}))\cdot\mathbf{B}\end{aligned} \tag{13}$$

where $\hat{\mathbf{K}} = \mathbf{K}\cdot(\mathbf{I}-\mathbf{W})\cdot\mathbf{K}$, $\mathbf{W} = \mathbf{diag}\,[\mathbf{w}_1, \cdots, \mathbf{w}_c]$ is an $N \times N$ block diagonal matrix, and $\mathbf{w}_i$ is a $C_i \times C_i$ matrix with terms all equal to: $\frac{1}{C_i}$.

We proceed by diagonalizing $\tilde{\mathbf{U}}^T\tilde{\mathbf{S}}_w\tilde{\mathbf{U}}$, a tractable matrix with size $m\times m$. Let $\tilde{\mathbf{p}}_i$ be the i-th eigenvector of $\tilde{\mathbf{U}}^T\tilde{\mathbf{S}}_w\tilde{\mathbf{U}}$, where $i = 1,\cdots,m$, sorted in **increasing** order of its corresponding eigenvalue $\tilde{\lambda}_i'$. In the set of ordered eigenvectors, those corresponding to the smallest eigenvalues minimize the denominator of Eq.7, and should be considered the most discriminative features. Let $\tilde{\mathbf{P}}_M = [\tilde{\mathbf{p}}_1,\cdots,\tilde{\mathbf{p}}_M]$ and $\tilde{\Lambda}_w = \mathbf{diag}[\tilde{\lambda}_1',\cdots,\tilde{\lambda}_M']$ be the selected $M(\leq m)$ eigenvectors and their corresponding eigenvalues, respectively. Then, the sought solution can be derived through $\tilde{\Gamma} = \tilde{\mathbf{U}}\tilde{\mathbf{P}}_M(\eta\mathbf{I}+\tilde{\Lambda}_w)^{-1/2}$, which is a set of optimal nonlinear discriminant feature bases.

4.4 Dimensionality Reduction and Feature Extraction

For any input pattern $\mathbf{z}$, its projection into the subspace spanned by the set of feature bases, $\tilde{\Gamma}$, derived in Section 4.3, can be computed by

$$\mathbf{y} = \tilde{\Gamma}^T\phi(\mathbf{z}) = \left(\tilde{\mathbf{E}}_m\cdot\tilde{\Lambda}_b^{-1/2}\cdot\tilde{\mathbf{P}}_M\cdot(\eta\mathbf{I}+\tilde{\Lambda}_w)^{-1/2}\right)^T\left(\tilde{\Phi}_b^T\phi(\mathbf{z})\right) \tag{14}$$

where $\tilde{\Phi}_b^T\phi(\mathbf{z}) = \left[\begin{array}{ccc}\tilde{\phi}_1 & \cdots & \tilde{\phi}_c\end{array}\right]^T\phi(\mathbf{z})$. We introduce an $(N\times 1)$ kernel vector,

$$\nu(\phi(\mathbf{z})) = \left[\begin{array}{ccccc}\phi_{11}^T\phi(\mathbf{z}) & \phi_{12}^T\phi(\mathbf{z}) & \cdots & \phi_{c(c_c-1)}^T\phi(\mathbf{z}) & \phi_{cc_c}^T\phi(\mathbf{z})\end{array}\right]^T, \tag{15}$$

which is obtained by dot products of $\phi(\mathbf{z})$ and each mapped training sample $\phi(\mathbf{z}_{ij})$ in $\mathbb{F}$. Reformulating Eq.14 by using the kernel vector, we obtain

$$\mathbf{y} = \Theta\cdot\nu(\phi(\mathbf{z})) \tag{16}$$

where

$$\Theta = \frac{1}{\sqrt{N}}\left(\tilde{\mathbf{E}}_m\cdot\tilde{\Lambda}_b^{-1/2}\cdot\tilde{\mathbf{P}}_M\cdot(\eta\mathbf{I}+\tilde{\Lambda}_w)^{-1/2}\right)^T\cdot\mathbf{B}\cdot\left(\mathbf{A}_{NC}^T - \frac{1}{N}\mathbf{1}_{NC}^T\right) \tag{17}$$

is an $(M\times N)$ matrix that can be computed off-line. Thus, through Eq.16, a low-dimensional nonlinear representation ($\mathbf{y}$) of $\mathbf{z}$ with enhanced discriminant power has been introduced. The detailed steps to implement the R-KDA method are summarized in Fig.3.

Input: A training set $\mathcal{Z}$ with C classes: $\mathcal{Z} = \{\mathcal{Z}_i\}_{i=1}^{C}$, each class containing $\mathcal{Z}_i = \{\mathbf{z}_{ij}\}_{j=1}^{C_i}$ examples, and the regularization parameter η.

Output: The matrix Θ; For an input example $\mathbf{z}$, its R-KDA based feature representation $\mathbf{y}$.

Algorithm:

Step 1. Compute the kernel matrix $\mathbf{K}$ using Eq.10.

Step 2. Compute $\tilde{\Phi}_b^T\tilde{\Phi}_b$ using Eq.11, and find $\tilde{\mathbf{E}}_m$ and $\tilde{\Lambda}_b$ from $\tilde{\Phi}_b^T\tilde{\Phi}_b$ in the way shown in Section 4.2.

Step 3. Compute $\tilde{\mathbf{U}}^T\tilde{\mathbf{S}}_w\tilde{\mathbf{U}}$ using Eq.12 and Eq.13, and find $\tilde{\mathbf{P}}_M$ and $\tilde{\Lambda}_w$ from $\tilde{\mathbf{U}}^T\tilde{\mathbf{S}}_w\tilde{\mathbf{U}}$ in the way depicted in Section 4.3;

Step 4. Compute Θ using Eq.17.

Step 5. Compute the kernel vector of the input $\mathbf{z}$, $\nu(\phi(\mathbf{z}))$, using Eq.15.

Step 6. The optimal nonlinear discriminant feature representation of $\mathbf{z}$ can be obtained by $\mathbf{y} = \Theta \cdot \nu(\phi(\mathbf{z}))$.

Fig. 3. R-KDA pseudo-code implementation (Matlab code is available by contacting the authors).

5 Comments

In this section ,we discuss the main properties and advantages of the proposed R-KDA method.

Firstly, R-KDA effectively deals with the SSS problem in the high-dimensional feature space by employing the regularized Fisher's criterion and the D-LDA subspace technique. It can be seen that R-KDA reduces to kernel YD-LDA and kernel JD-LDA (also called KDDA [11]) when $\eta = 0$ and $\eta = 1$, respectively. Varying the values of η within $[0, 1]$ leads to a set of intermediate kernel D-LDA variants between kernel YD-LDA and KDDA. Since the subspace spanned by $\tilde{\Psi}$ may contain the intersection space ($\mathbf{A}' \cap \mathbf{B}$), it is possible that there exist zero or very small eigenvalues in $\tilde{\Lambda}_w$, which have been shown to be high variance for estimation in the SSS environments [7]. As a result, any bias arising from the eigenvectors corresponding to these eigenvalues is dramatically exaggerated due to the normalization process ($\tilde{\mathbf{P}}_M\tilde{\Lambda}_w^{-1/2}$). Against the effect, the introduction of the regularization helps to decrease the importance of these highly unstable eigenvectors, thereby reducing the overall variance. Also, there may exist the zero eigenvalues in $\tilde{\Lambda}_w$, which are used as divisors in YD-LDA due to $\eta = 0$. However, it is not difficult to see that the problem can be avoided in the R-KDA solution, $\tilde{\Psi} = \tilde{\mathbf{U}}\tilde{\mathbf{P}}_M(\eta\mathbf{I} + \tilde{\Lambda}_w)^{-1/2}$, simply by setting the parameter $\eta > 0$. In this way, R-KDA can exactly extract the optimal discriminant features from both inside and outside of $\tilde{\mathbf{S}}_w$'s null space, while avoiding the risk of experiencing high variance in estimating the scatter matrices at the same time. This point makes R-KDA significantly

different from existing nonlinear discriminant analysis methods such as GDA in the SSS situations.

In GDA, to remove the null space of $\tilde{\mathbf{S}}_w$, it is required to compute the pseudo inverse of the kernel matrix $\mathbf{K}$, which could be extremely ill-conditioned when certain kernels or kernel parameters are used. Pseudo inversion is based on inversion of the nonzero eigenvalues. Due to round-off errors, it is not easy to identify the true null eigenvalues. As a result, numerical stability problems often occur [22]. However, it can be seen from the derivation of R-KDA that such problems are avoided in R-KDA. The improvement can be observed also in experimental results reported in Figs.8-9:**Left**.

In GDA, both the two eigen-decompositions of $\tilde{\mathbf{S}}_b$ and $\tilde{\mathbf{S}}_w$ have to be implemented in the feature space $\mathbb{F}$. In contrast with this, it can be seen from Section 4.3 that the eigen-decomposition of $\tilde{\mathbf{S}}_w$ is replaced by that of $\tilde{\mathbf{U}}^T\tilde{\mathbf{S}}_w\tilde{\mathbf{U}}$, which is an $(m \times m)$ matrix with $m \leq C-1$. Also, it should be noted at this point that it generally requires much more computational costs to implement an eigen-decomposition for $\tilde{\mathbf{S}}_w$ than $\tilde{\mathbf{S}}_b$, due to $C \ll N$ in most cases. Therefore, based on the two factors, it is not difficult to see that the computational complexity of R-KDA is significantly reduced compared to GDA. This point is demonstrated by the face recognition experiment reported in Section 6.2, where R-KDA is approximately 20 times faster than GDA.

6 Experimental Results

Two sets of experiments are included here to illustrate the effectiveness of the R-KDA method in different learning scenarios. The first experiment is conducted on Fisher's iris data [6] to assess the performance of R-KDA in traditional large-sample-size situations. Then, R-KDA is applied to face recognition tasks in the second experiment, where various SSS settings are introduced. In addition to R-KDA, other two kernel-based feature extraction methods, KPCA and GDA, are implemented to provide a comparison of performance, in terms of classification error and computational cost.

6.1 Fisher's Iris Data

The iris flower data set originally comes from Fisher's work [6]. The set consists of $N = 150$ iris specimens of $C = 3$ species (classes). Each specimen is represented by a four-dimensional vector, describing four parameters, sepal length/width and petal length/width. Among the three classes, one is linearly separable from the other two, while the latter are not linearly separable from each other. Due to $J(= 4) \ll N$, there is no SSS problem introduced in this case, and thus we set $\eta = 0.001$ for R-KDA.

Firstly, it is of interest to observe how R-KDA linearizes and simplifies the complicated data distribution as GDA did in [2]. To this end, four types of feature bases are generalized from the iris set by utilizing the LDA, KPCA,

R-KDA and GDA algorithms, respectively. These feature bases form four subspaces, accordingly. Then, all the examples are projected to the four subspaces. For each example, its projections in the first two most significant feature bases of each subspace are visualized in Fig.4. As analyzed in Section 2.3, the PCA-based features are optimized with focus on object reconstruction. Not surprisingly, it can be seen from Fig.4 that the subjects are not separable in the KPCA subspace, even with the introduction of nonlinear kernel. Unlike the PCA approaches, LDA optimizes the feature representation based on separability criteria. However, subject to the limitation of linearity, the two non-separable classes remain non-separable in the LDA subspace. In contrast to this, we can see the linearization property in the R-KDA and GDA subspaces, where all of classes are well linearly separable when a RBF kernel with appropriate parameters is used.

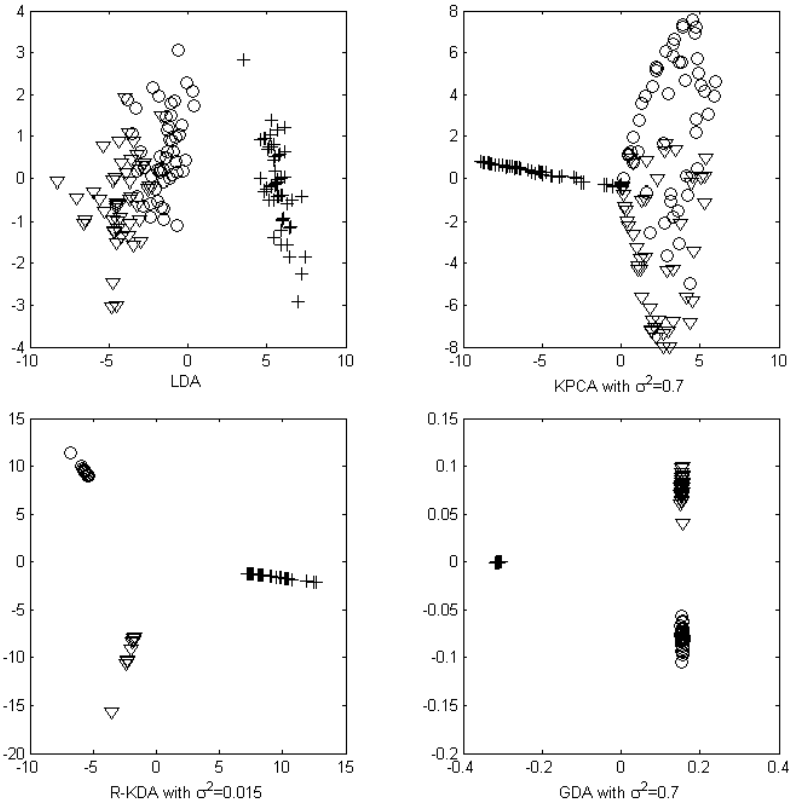

Fig. 4. Iris data are project to four feature spaces obtained by LDA, KPCA, R-KDA and GDA respectively. LDA is derived from R-KDA by using a polynomial kernel with degree one, while all other three kernel methods use a RBF kernel.

Also, we examine the classification error rate (CER) of the three kernel feature extraction algorithms compared here with the so-called "leave one out" test method. Following the recommendation in [2], a RBF kernel with $\sigma^2 = 0.7$ is used for all these algorithms in this experiment. The CERs obtained by GDA and R-KDA are only 7.33% and 6% respectively, while the CER of KPCA with the same feature number ($M = 2$) to the formers goes up to 20%. The two experiments conducted on the iris data indicate that the performance of R-KDA is comparable to that of GDA in the large-sample-size learning scenarios, although the former is designed specifically to address the SSS problem.

6.2 Face Recognition

Face Recognition Evaluation Design

Face recognition is one of current most challenging applications in the pattern recognition literature [4, 23, 30, 31, 35]. In this work, the algorithms are evaluated with two widely used face databases, UMIST [8] and FERET [19]. The UMIST repository is a multi-view database, consisting of 575 images of 20 people, each covering a wide range of poses from profile to frontal views [8]. The FERET database has been considered current most comprehensive and representative face database [19, 20]. For the convenience of preprocessing, we only choose a medium-size subset of the database. The subset consists of 1147 images of 120 people, each one having at least 6 samples so that we can generalize a set of SSS learning tasks. These images cover a wide range of variations in illumination and facial expression/details with pose angles less than 30 degrees. Figs.5-6 depict some examples from the two databases. For computational convenience, each image is represented as a column vector of length $J = 10304$ for UMIST and $J = 17154$ for FERET.

Fig. 5. Some samples of four people come from the UMIST database.

The SSS problem is defined in terms of the number of available training samples per subject, L. Thus the value of L has a significant influence on

Fig. 6. Some samples of eight people come from the normalized FERET database.

the required strength of regularization. To study the sensitivity of the performance, in terms of *correct recognition rate* (CRR), to L, five tests were performed with various L values ranging from $L = 2$ to $L = 6$. For a particular L, any database evaluated here is randomly partitioned into two subsets: a training set and a test set. The training set is composed of $(L \times C)$ samples: L images per person were randomly chosen. The remaining $(N - L \times C)$ images were used to form the test set. There is no overlapping between the two subsets. To enhance the accuracy of the assessment, five runs of such a partition were executed, and all the results reported below have been averaged over the five runs.

CRRs with Varying Regularization Parameter

In this experiment, we examine the performance of R-KDA with varying regularization parameter values in different SSS scenarios, $L = 2 \sim 4$. For simplicity, R-KDA is only tested with a linear polynomial kernel in the FERET subset. Fig.7 depicts the obtained CRRs as a function of (M, η), where M is the number of feature vectors used.

The parameter η controls the strength of regularization, which balances the tradeoff between variance and bias in the estimation for the zero or small eigenvalues of the within-class scatter matrix. Varying the η values within $[0, 1]$ leads to a set of intermediate kernel D-LDA variants between kernel YD-LDA and KDDA. In theory, kernel YD-LDA with no bias introduced should be the best performer among these variants if sufficient training samples are available. It can be observed at this point from Fig.7 that the CRR peaks gradually moved from the right side toward the left side ($\eta = 0$) that is the case of kernel YD-LDA as L increases. Small values of η have been good enough for the regularization requirement in many cases ($L \geq 4$) as shown in Fig.7. However, it also can be seen that kernel YD-LDA performed poorly when $L = 2, 3$. This should be attributed to the high variance in the estimate of $\tilde{\mathbf{S}}_w$ due to insufficient training samples. In these cases, even $\tilde{\mathbf{U}}^T \tilde{\mathbf{S}}_w \tilde{\mathbf{U}}$ is

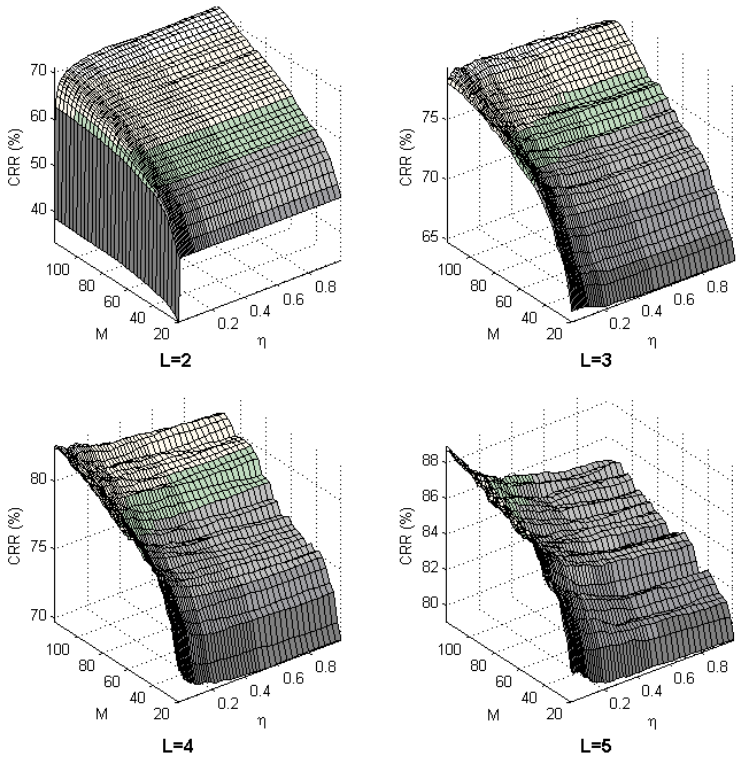

Fig. 7. CRRs(M, η) obtained by R-KDA with a linear polynomial kernel.

singular or close to singular, and the resulting effect is to dramatically exaggerate the importance associated with the eigenvectors corresponding to the smallest eigenvalues. Against the effect, the introduction of regularization helps to decrease the larger eigenvalues and increase the smaller ones, thereby counteracting for some extent the bias. This is also why KDDA outperforms kernel YD-LDA when L is small.

Performance Comparison with KPCA and GDA

This experiment compares the performance of the R-KDA algorithms, in terms of the CRR and the computational cost, to the KPCA and GDA algorithms. For simplicity, only the RBF kernel is tested in this work, and the classification is performed with the nearest neighbor rule.

Tables 2-3 depict a quantitative comparison of the best CRRs with corresponding parameter values (σ^{2*}, M^*), found by the three methods in the UMIST and FERET databases, each one having introduced five SSS cases from $L = 2$ to $L = 6$. In addition to σ^2 and M, R-KDA's performance is

Table 2. Comparison of the best found CRRs (%) with corresponding parameter values in the UMIST database.

Methods	KPCA			GDA			R-KDA			
	CRR	σ^{2*}	M^*	CRR	σ^*	M^*	CRR	σ^*	M^*	η
$L=2$	57.91	2.11×10^7	34	62.92	1.34×10^8	19	66.73	1.5×10^8	14	1.0
$L=3$	69.67	5.33×10^7	58	76.00	3.72×10^7	18	80.97	1.5×10^8	14	0.001
$L=4$	78.02	6.94×10^7	78	84.20	5.33×10^7	19	89.17	1.5×10^8	11	0.001
$L=5$	84.67	2.11×10^7	95	90.32	5.33×10^7	19	93.01	1.34×10^8	13	0.001
$L=6$	87.91	6.94×10^7	119	92.97	6.94×10^7	19	95.30	1.5×10^8	14	0.001

Table 3. Comparison of the best found CRRs (%) with corresponding parameter values in the FERET database.

Methods	KPCA			GDA			R-KDA			
	CRR	σ^{2*}	M^*	CRR	σ^*	M^*	CRR	σ^*	M^*	η
$L=2$	60.93	2.34×10^5	238	71.18	2.68×10^4	118	73.38	3.0×10^5	102	1.0
$L=3$	67.32	7.44×10^3	358	80.58	2.68×10^4	118	85.51	3.0×10^5	106	0.001
$L=4$	71.39	2.34×10^5	468	85.07	2.68×10^4	118	88.34	3.0×10^5	108	0.001
$L=5$	75.32	2.03×10^4	590	88.48	2.68×10^4	118	91.96	2.34×10^5	104	0.001
$L=6$	77.85	2.03×10^4	716	90.21	2.03×10^4	118	92.74	3.0×10^5	110	0.001

affected by the regularization parameter, η. Considering the high computational cost of searching the best η^*, we simply set $\eta = 1.0$ for the $L = 2$ cases and $\eta = 0.001$ for other cases based on the observation and analysis of the results in Section 6.2. Also, the CRRs as a function of σ^2 and M respectively in several representative UMIST cases are shown in Figs.8-9. From these results, it can be seen that R-KDA is the top performer in all the experimental cases. On average, R-KDA leads KPCA and GDA up to 9.4% and 3.8% in the UMIST database, and 15.8% and 3.3% in the FERET database. It should be also noted that Figs.8-9:**Left** reveal the numerical stability problems existing in practical implementations of GDA. Comparing GDA to R-KDA, we can see that the later is more stable and predictable, resulting in a cost-effective determination of parameter values during the training phase.

In addition to the CRR, it is of interest to compare the performance with respect to the computational complexity. For each of the methods evaluated here, the simulation process consists of 1) a training stage that includes all operations performed in the training set; 2) a test stage for the CRR determination. The computational times consumed by these methods with the parameter configuration depicted in Tables 2-3 are reported in Table 4. T_{trn} and T_{tst} are the amounts of time spent on training and testing respectively. The simulation studies reported in this work were implemented on a personal computer system equipped with a 2.0GHz Intel Pentium 4 processor and 1.0 GB RAM. All programs are written in Matlab v6.5 and executed in MS Windows

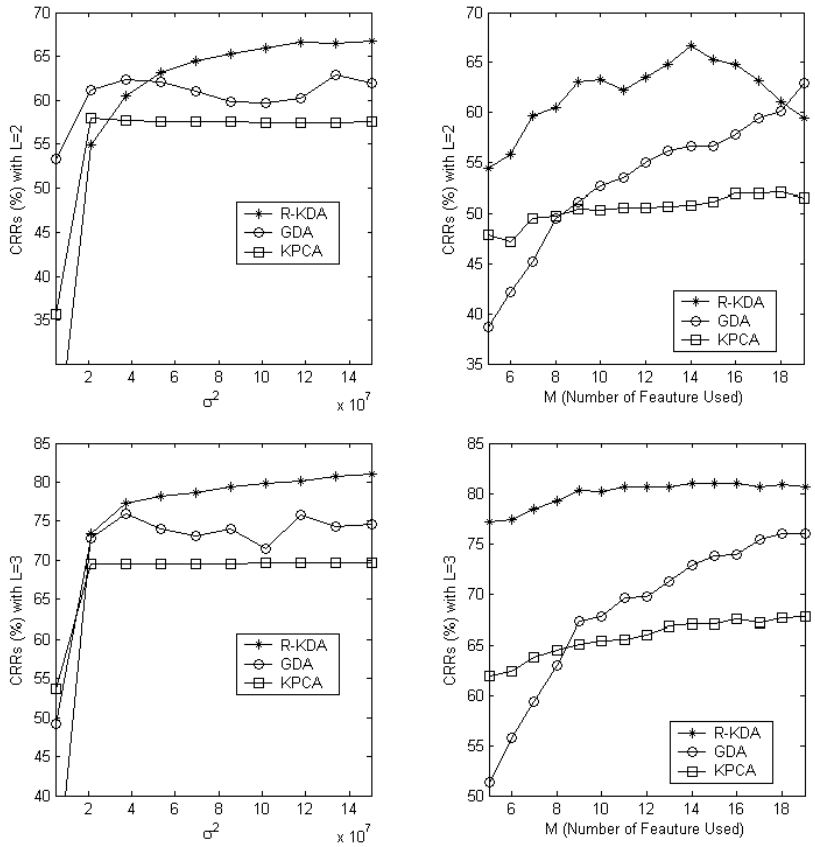

Fig. 8. A comparison of CRRs based on the RBF kernel function in the UMIST cases of $L = 2 \sim 3$. **Left**: CRRs as a function of σ^2 with the best found M^*. **Right**: CRRs as a function of M with the best found σ^{2*}.

2000. For the convenience of comparison, we introduce a quantitative statistic in Table 5 regarding the computational time of KPCA or GDA over that of R-KDA, $\xi_{trn}(\cdot) = T_{trn}(\cdot)/T_{trn}(\text{R-KDA})$ and $\xi_{tst}(\cdot) = T_{tst}(\cdot)/T_{tst}(\text{R-KDA})$. As analyzed in Section 5, the computational cost of R-KDA should be less than that of GDA. It can be observed clearly at this point from Table 5 that R-KDA is approximately 20 times faster than GDA in both the training and test phases. Moreover, R-KDA is more than 3 times in training and 4 times in testing faster than KPCA. The higher computational complexity of KPCA is due to the significantly larger feature number used, M^* as shown in Tables 2-3. The advantage of R-KDA in computation is particularly important for the practical face recognition tasks, where algorithms are often required to deal with huge scale databases.

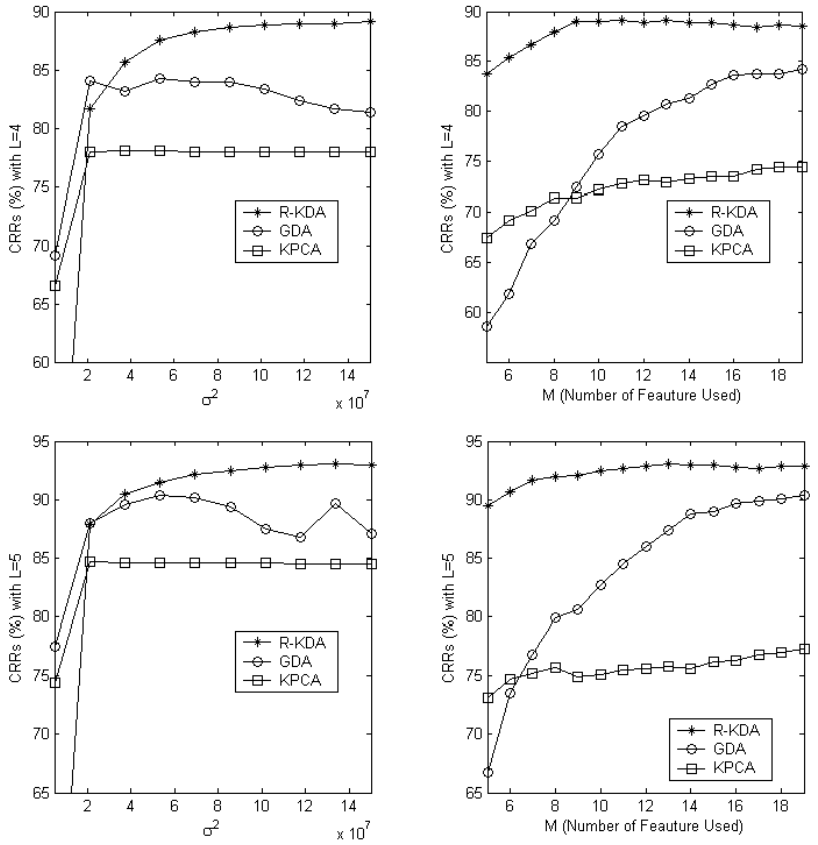

Fig. 9. A comparison of CRRs based on the RBF kernel function in the UMIST cases of $L = 4 \sim 5$. **Left**: CRRs as a function of σ^2 with the best found M^*. **Right**: CRRs as a function of M with the best found σ^{2*}.

Table 4. A comparison of computational times, $T_{trn} + T_{tst}$ (Seconds).

DBS	Methods	$L = 2$	$L = 3$	$L = 4$	$L = 5$	$L = 6$
UMIST	KPCA	0.8+11.3	2.1+12.0	4.5+16.9	7.3+25.2	8.5+19.5
	GDA	6.2+44.9	14.2+65.1	25.7+83.9	40.7+101.1	55.9+109.0
	R-KDA	0.3+2.2	0.7+3.2	1.2+4.0	2.0+4.7	2.8+5.4
FERET	KPCA	76+203	134+205	320+323	375+254	526+245
	GDA	392+750	905+1014	1641+1156	2662+1198	3861+1121
	R-KDA	19+38	42+50	76+58	117+57	170+57

7 Conclusion

Due to the extremely high dimensionality of the kernel feature spaces, the SSS problem is often encountered when traditional kernel discriminant anal-

Table 5. A comparison of the computational time of KPCA or GDA over that of R-KDA, $\xi_{trn} + \xi_{tst}$.

DBS	Methods	$L = 2$	$L = 3$	$L = 4$	$L = 5$	$L = 6$	Aver.
UMIST	KPCA	2.6+5.1	2.9+3.8	3.6+4.2	3.8+5.4	3.0+3.6	3.2+4.4
	GDA	19.4+20.2	20.0+20.4	20.6+21.1	20.8+21.5	20.1+20.0	20.2+20.6
FERET	KPCA	4.0+5.3	3.2+4.1	4.2+5.5	3.2+4.4	3.1+4.3	3.5+4.7
	GDA	20.8+19.6	21.5+20.2	21.6+19.8	22.7+20.9	22.7+19.6	21.9+20.0

ysis methods are applied to many practical tasks such as face recognition. To address the problem, a regularized kernel discriminant analysis method is introduced in this chapter. The proposed method is based a novel regularized Fisher's discriminant criterion, which is particularly robust against the SSS problem compared to the original one used in traditional linear/kernel discriminant analysis methods. It has been also shown that a series of traditional LDA variants and their kernel versions including the recently introduced YD-LDA, JD-LDA and KDDA can be derived from the proposed framework by adjusting the regularization and kernel parameters. Experimental results obtained in the face recognition tasks indicate that the CRR performance of the proposed R-KDA algorithm is overall superior to those obtained by the KPCA or GDA approaches in various SSS situations. Also, the R-KDA method has significantly less computational complexity than the GDA method. This point has been demonstrated in the face recognition experiments, where R-KDA is approximately 20 times faster than GDA in both the training and test phases.

In conclusion, the R-KDA algorithm provides a general pattern recognition framework for nonlinear feature extraction from high-dimensional input patterns in the SSS situations. We expect that in addition to face recognition, R-KDA will provide excellent performance in applications where classification tasks are routinely performed, such as content-based image indexing and retrieval, video and audio classification.

Acknowledgments

Portions of the research in this dissertation use the FERET database of facial images collected under the FERET program [19]. We would like to thank the FERET Technical Agent, the U.S. National Institute of Standards and Technology (NIST) for providing the FERET database. Also, We would like to thank Dr. Daniel Graham and Dr. Nigel Allinson for providing the UMIST face database [8].

References

1. Aizerman, M. A., Braverman, E. M., Rozonoér, L. I. (1964) Theoretical foundations of the potential function method in pattern recognition learning. *Au-*

tomation and Remote Control, 25:821–837.
2. Baudat, G., Anouar, F. (2000) Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12:2385–2404.
3. Belhumeur, P. N., Hespanha, J. P., Kriegman, D. J. (1997) Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720.
4. Chellappa, R., Wilson, C., Sirohey, S. (1995) Human and machine recognition of faces: A survey. *The Proceedings of the IEEE*, 83:705–740.
5. Chen, L.-F., Liao, H.-Y. M., Ko, M.-T., Lin, J.-C., Yu., G.-J. (2000) A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, 33:1713–1726.
6. Fisher, R. (1936) The use of multiple measures in taxonomic problems. *Ann. Eugenics*, 7:179–188.
7. Friedman, J. H. (1989) Regularized discriminant analysis. *Journal of the American Statistical Association*, 84:165–175.
8. Graham, D. B., Allinson, N. M. (1998) Characterizing virtual eigensignatures for general purpose face recognition. In Wechsler, H., Phillips, P. J., Bruce, V., Soulie, F. F., Huang, T. S., editors, *Face Recognition: From Theory to Applications, NATO ASI Series F, Computer and Systems Sciences*, 163:446–456.
9. Kanal, L., Chandrasekaran, B. (1971) On dimensionality and sample size in statistical pattern classification. *Pattern Recognition*, 3:238–255.
10. Liu, K., Cheng, Y., Yang, J., Liu, X. (1992) An efficient algorithm for foley-sammon optimal set of discriminant vectors by algebraic method. *Int. J. Pattern Recog. Artif. Intell.*, 6:817–829.
11. Lu, J., Plataniotis, K., Venetsanopoulos, A. (2003) Face recognition using kernel direct discriminant analysis algorithms. *IEEE Transactions on Neural Networks*, 14(1):117–126.
12. Lu, J., Plataniotis, K., Venetsanopoulos, A. (2003) Face recognition using LDA based algorithms. *IEEE Transactions on Neural Networks*, 14(1):195–200.
13. Lu, J., Plataniotis, K., Venetsanopoulos, A. (2003) Regularized discriminant analysis for the small sample size problem in face recognition. *Pattern Recognition Letter*, 24(16):3079–3087, December.
14. Lu, J., Plataniotis, K., Venetsanopoulos, A. (in press) Regularization studies of linear discriminant analysis in small sample size scenarios with application to face recognition. *Pattern Recognition Letter*, Accepted in September 2004.
15. Martínez, A. M., Kak, A. C. (2001) PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233.
16. McLachlan, G. (1992) *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, New York.
17. Mercer, J. (1909) Functions of positive and negative type and their connection with the theory of integral equations. *Philos. Trans. Roy. Soc. London*, A 209:415–446.
18. Müller, K. R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B. (2001) An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, March.
19. Phillips, P. J., Moon, H., Rizvi, S. A., Rauss, P. J. (2000) The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104.

20. Phillips, P. J., Wechsler, H., Huang, J., Rauss, P. (1998) The FERET database and evaluation procedure for face recognition algorithms. *Image and Vision Computing J*, 16(5):295–306.
21. Raudys, S. J., Jain, A. K. (1991) Small sample size effects in statistical pattern recognition: Recommendations for practitioners. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):252–264.
22. Ruiz, A., Teruel, P. L. de. (2001) Nonlinear kernel-based statistical pattern analysis. *IEEE Transactions on Neural Networks*, 12(1):16–32, January.
23. Samal, A., Iyengar, P. A. (1992) Automatic recognition and analysis of human faces and facial expressions: A survey. *Pattern Recognition*, 25:65–77.
24. Schölkopf, B. (1997) *Support Vector Learning*. Oldenbourg-Verlag, Munich, Germany.
25. Schölkopf, B., Burges, C., Smola, A. J. (1999) *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, MA.
26. Schölkopf, B., Smola, A., Müller, K. R. (1999) Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319.
27. Schölkopf, B., Smola, A. J. (2001) *Learning with Kernels*. MA: MIT Press, Cambridge.
28. Smola, A. J., Schölkopf, B., Müller, K. R. (1998) The connection between regularization operators and support vector kernels. *Neural Networks*, 11:637–649.
29. Swets, D. L., Weng, J. (1996) Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:831–836.
30. Turk, M. (2001) A random walk through eigenspace. *IEICE Trans. Inf. & Syst.*, E84-D(12):1586–1695, December.
31. Valentin, D., Alice, H. A., Toole, J. O., Cottrell, G. W. (1994) Connectionist models of face processing: A survey. *Pattern Recognition*, 27(9):1209–1230.
32. Vapnik, V. N. (1995) *The Nature of Statistical Learning Theory*. Springer-Verlag, New York.
33. Wald, P., Kronmal, R. (1977) Discriminant functions when covariance are unequal and sample sizes are moderate. *Biometrics*, 33:479–484.
34. Yu, H., Yang, J. (2001) A direct LDA algorithm for high-dimensional data - with application to face recognition. *Pattern Recognition*, 34:2067–2070.
35. Zhao, W., Chellappa, R., Phillips, P., Rosenfeld, A. (2003) Face recognition: A literature survey. *ACM Computing Surveys*, 35(4):399–458, December.
36. Zien, A., Rätsch, G., Mika, S., Schölkopf, B., Lengauer, T., Müller, K.-R. (2000) Engineering support vector machine kernels that recognize translation initiation sites in dna. *Bioinformatics*, 16:799–807.