KLDA - AN ITERATIVE APPROACH TO FISHER DISCRIMINANT ANALYSIS

Fangfang Lu[1], *Hongdong Li*[1,2]

[1] Research School of Information Sciences and Engineering, Australian National University
[2] Vision Science, Technology and Applications Program, National ICT Australia*

ABSTRACT

In this paper, we present an iterative approach to Fisher discriminant analysis called Kullback-Leibler discriminant analysis (KLDA) for both linear and nonlinear feature extraction. We pose the conventional problem of discriminative feature extraction into the setting of function optimization and recover the feature transformation matrix via maximization of the objective function. The proposed objective function is defined by pairwise distances between all pairs of classes and the Kullback-Leibler divergence is adopted to measure the disparity between the distributions of each pair of classes. Our proposed algorithm can be naturally extended to handle nonlinear data by exploiting the kernel trick. Experimental results on the real world databases demonstrate the effectiveness of both the linear and kernel versions of our algorithm.

Index Terms— Linear Discriminant Analysis, Kernel Fisher Discriminant Analysis, Kullback-Leibler Divergence, Optimization

1. INTRODUCTION

Discriminant Analysis methods have been studied for many years within the pattern recognition community. The aim is to find a transformation of the input data to a lower dimensional subspace which best discriminate between different classes.

Conventional Linear Discriminant Analysis (LDA) works by simultaneously maximizing the between-class scatter and minimizing the within-class scatter for the transformed data[1]. The use of scatter matrices offers much computational advantage by reducing the original problem to generalized eigenvalue decomposition, but conceptually, it is quite restricted in the modeling power. First, LDA assumes each class satisfy a Gaussian distribution with equal covariance matrices, so that a unique within-class scatter matrix can be used to represent the average variance of data distribution within each class. This is, however, a very strong assumption, as real-world data are very complicated and different classes may have different distributions. Second, LDA is optimal for binary classification [2]. For multiclass classification, it uses single within- and between-class scatter matrices to represent the variations within and between different classes without considering the pairwise relationships between any two classes. The resulting transformation may overemphasize well-separated classes but is not optimized to distinguish between overlapping classes which are relatively harder to classify and should be paid more attention.

Various extensions of LDA [3, 2, 4] have been proposed to handle the above two problems. Heteroscadastic Discriminant Analysis (HDA) was proposed in the previous study on LDA [3, 4], which aims at relaxing the equal covariance constraint and modeling the variability in class conditional distributions. Pairwise discriminant analysis [2] was proposed to remedy the class-balance problem by employing an objective function defined for all pairs of classes. However, most of these methods still much rely on the scatter matrices and an equal within-class scatter matrix was used anyway.

In this paper, we proposed a new approach, called KLDA, to pairwise heteroscadastic discriminant analysis. Instead of using within- and between-class scatter matrices to measure the separability between different classes, we used the information theoretic Kullback-Leibler (KL) divergence measure as an indicator for class separability [5]. We then treat Fisher discriminant analysis as an optimization problem, where the objective functionis derived by accumulating the divergence values for all pairs of two classes. The optimal feature transformation matrix is attained at the local maximum of the objective function through an iterative optimization process.In contrast to previous approach which utilizes Chernoff distance to rectify the between-class scatter matrix rather than employing it as the class separability measure [4], our approach maximizes pairwise class separability directly in terms of the sum of KL divergences, and hence is theoretically optimal. Moreover, KL divergence also has a simpler partial differential term with respect to the transformation matrix compared to the Chernoff distance, which makes the iterative update process computationally more efficient. Our algorithm can also be generalized to the kernel version to handle linearly inseparable data via the kernel trick.

The remainder of this paper is organized as follows. Section 2 presents the main algorithm of Kullback-Leibler Discriminant Analysis, including both linear and kernel KLDA. Section 3 presents experimental results on two real-world databases on image labeling and character recognition. Conclusions are given in the last section.

*National ICT Australia is funded through the Australian Government's Backing Australia's Ability Initiative, in part through the ARC.

2. KULLBACK-LEIBLER DISCRIMINANT ANALYSIS (KLDA)

2.1. Preliminaries

LDA We first define the generic problem of linear feature extraction for classification treated in this paper. Given the training data $\mathbf{x}_i \in \mathcal{R}^D$ $(i = 1, \ldots, N)$ the purpose is to find a transformation matrix $\mathbf{A} \in \mathcal{R}^{D \times d}$ that projects the input vector $\mathbf{x}_i$ to the point $\mathbf{y}_i = \mathbf{A}^T \mathbf{x}_i$ in a lower dimensional feature space $\mathcal{R}^d (d \ll D)$ so as to maximize certain optimality criterion. This is often posed as an optimization function over the transformation matrix $\mathbf{A}$. LDA is one such method which minimizes the following cost function,

$$\min f(\mathbf{A}) = tr((\mathbf{A}^T \mathbf{S}_w \mathbf{A})^{-1} (\mathbf{A}^T \mathbf{S}_b \mathbf{A})) \quad (1)$$

$$\mathbf{S}_w = \sum_{j=1}^{c} \sum_{\mathbf{x}_i \in C_j} (\mathbf{x}_i - \mu_j)(\mathbf{x}_i - \mu_j)^T$$

$$\mathbf{S}_b = \sum_{j=1}^{c} n_j (\mu_j - \mu)(\mu_j - \mu)^T$$

where c is the number of different classes, C_j denotes the set of samples in class j, μ_j and μ are the mean of class j and the sample mean respectively, and n_j is the size of class j. $\mathbf{S}_w$ and $\mathbf{S}_b$ represent the within and between-class scatter matrices respectively. They can be viewed as the average class-specific covariance and mean distance over all the classes. The purpose of Equation 1 is to maximize the between-class scatter while preserving within-class dispersion in the transformed feature space. Its solution can be obtained by solving the generalized eigenvalue problem $\mathbf{S}_b \mathbf{A} = \lambda \mathbf{S}_w \mathbf{A}$ and taking the eigenvectors corresponding to the leading eigenvalues of $\mathbf{S}_w^{-1} \mathbf{S}_b$. In the implementation of LDA, a regularization parameter is usually added to the diagonal elements of $\mathbf{S}_w$ for numerical stability. This is especially important for small sample size problems where S_w is near singular.

KFDA KFDA is a nonlinear extension to LDA [6]. The idea is to map the input data to a higher-dimensional nonlinear feature space as denoted by $\phi : \mathcal{R}^D \rightarrow \mathcal{H}$, called the Reproducing Kernel Hilbert Space (RKHS), and then perform LDA in the feature space instead. Though the explicit form of mapping is unknown, the inner product in the RKHS can be represented in closed form by the kernel function defined over input vectors $k(\mathbf{x}, \mathbf{y}) =< \phi(\mathbf{x}), \phi(\mathbf{y}) >$. Moreover, the projected samples $\phi(\mathbf{x}_i)$ form a set of bases for the RKHS, and any vector in the RKHS can be represented by the linear combination of these bases. Hence the projection from the RKHS to the output space is given by $\mathbf{W} = \mathbf{\Phi}(\mathbf{X})\mathbf{A} = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_N)]\mathbf{A}$, where $\mathbf{A}$ is the coefficient matrix for the linear combination. Then KFDA can be formulated as the following optimization problem,

$$\begin{aligned} \min f(\mathbf{A}) &= tr((\mathbf{A}^T \mathbf{\Phi}(\mathbf{X})^T \mathbf{\Phi}(\mathbf{X}) \mathbf{S}_w \mathbf{\Phi}(\mathbf{X})^T \mathbf{\Phi}(\mathbf{X}) \mathbf{A})^{-1} \\ &\quad (\mathbf{A}^T \mathbf{\Phi}(\mathbf{X})^T \mathbf{\Phi}(\mathbf{X}) \mathbf{S}_b \mathbf{\Phi}(\mathbf{X})^T \mathbf{\Phi}(\mathbf{X}) \mathbf{A})) \\ &= tr((\mathbf{A}^T \mathbf{K} \mathbf{S}_w \mathbf{K} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{K} \mathbf{S}_b \mathbf{K} \mathbf{A}) \end{aligned} \quad (2)$$

$$\mathbf{S}_w = \sum_{j=1}^{c} (I_d - \mathbf{e}_j \mathbf{e}_j^T / n_j)(I_d - \mathbf{e}_j \mathbf{e}_j^T / n_j)^T$$

$$\mathbf{S}_b = \sum_{j=1}^{c} (\mathbf{e}_j 1_{n_j}^T - 1_N 1_{n_j}^T)(\mathbf{e}_j 1_{n_j}^T - 1_N 1_{n_j}^T)^T / n_j$$

where I_d is the identity matrix, e_j is the indicator vector for class j such that $e_j(i) = 1$ if $\mathbf{x}_i \in C_j$ and otherwise $e_j(i) = 0$, 1_n denotes a column vector of length n with straight 1s. $\mathbf{K}$ is the $N \times N$ Gram matrix whose (i, j)th entry is given by the kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$. It can be clearly seen that the cost function of KFDA as defined above share a similar form as the cost function for LDA defined in Equation 1. Hence the transformation matrix $\mathbf{A}$ can be obtained, again, by solving a generalized eigenvalue decomposition problem $\mathbf{K}\mathbf{S}_b\mathbf{K}\mathbf{A} = \lambda \mathbf{K}\mathbf{S}_w\mathbf{K}\mathbf{A}$ and taking the leading eigenvectors.

2.2. Linear KLDA

In this section, we define our new objective function for doing discriminant analysis in a pairwise fashion. We make use of the KL-divergence as a measure of separability for any two classes by viewing them as probabilistic distributions of the training data. The objective function is defined as the sum of KL-divergences between all pairs of classes as follows

$$\max f(\mathbf{A}) = \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} w_i w_j KL(p_i(\mathbf{A}^T \mathbf{x}), p_j(\mathbf{A}^T \mathbf{x})) \quad (3)$$

where $p_i(\mathbf{A}^T x)$ and $p_j(\mathbf{A}^T x)$ denotes the sample distributions of classes i and j after linear transformation by matrix $\mathbf{A}$, $w_i \infty n_i$ and $w_j \infty n_j$ are prior probability of classes i and j to balance between different sized classes. Then the optimal transformation matrix $\mathbf{A}$ can be found at the maximum of the objective function defined above.

The original KL divergence as first proposed in [5] is an asymmetric distance measure. Here we adopt a symmetric version of it for measuring the disparity between two distributions $p_i(\mathbf{x})$ and $p_j(\mathbf{x})$, which is defined as follows

$$KL(p_i(\mathbf{x}), p_j(\mathbf{x})) = \int (p_i(\mathbf{x}) - p_j(\mathbf{x})) \log \frac{p_i(\mathbf{x})}{p_j(\mathbf{x})} d\mathbf{x} \quad (4)$$

Assume both $p_i(\mathbf{x})$ and $p_j(\mathbf{x})$ satisfy Gaussian distributions, the KL divergence can be expressed in closed form

$$\begin{aligned} KL(p_i(\mathbf{x}), p_j(\mathbf{x})) &= (\mu_i - \mu_j)^T (\mathbf{\Sigma}_i^{-1} + \mathbf{\Sigma}_j^{-1})(\mu_i - \mu_j) \\ &\quad + tr(\mathbf{\Sigma}_i^{-1} \mathbf{\Sigma}_j + \mathbf{\Sigma}_j^{-1} \mathbf{\Sigma}_i - 2I_d) \end{aligned} \quad (5)$$

where μ_i, μ_j are the mean vectors of Gaussian distributions for $p_i(\mathbf{x})$ and $p_j(\mathbf{x})$, $\mathbf{\Sigma}_i$ and $\mathbf{\Sigma}_j$ are the covariance matrices. Both the mean vectors and covariances matrices can be estimated using sample mean and sample covariance from the training data.

$$\hat{\mu}_j = \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i / n_j \quad (6)$$

$$\hat{\mathbf{\Sigma}}_j = \sum_{\mathbf{x}_i \in C_j} (\mathbf{x}_i - \mu_j)(\mathbf{x}_i - \mu_j)^T / n_j + \lambda \mathbf{I}_d \quad (7)$$

where λ is the regularization parameter added to make the matrix inverse operation numerically more stable.

With the transformation matrix $\mathbf{A}$, the mean $\hat{\mu}$ and the covariance matrix $\hat{\mathbf{\Sigma}}$ convert to $\mathbf{A}^T\hat{\mu}$ and $\mathbf{A}^T\hat{\mathbf{\Sigma}}\mathbf{A}$ in the transformed feature space. The KL divergence for the transformation is thus given by

$$\begin{aligned} kl_{i,j} &= \hat{\mu}_{i,j}^T \mathbf{A}((\mathbf{A}^T\hat{\mathbf{\Sigma}}_i\mathbf{A})^{-1} + (\mathbf{A}^T\hat{\mathbf{\Sigma}}_j\mathbf{A})^{-1})\mathbf{A}^T\hat{\mu}_{i,j} \quad (8) \\ &+tr((\mathbf{A}^T\hat{\mathbf{\Sigma}}_i\mathbf{A})^{-1}(\mathbf{A}^T\hat{\mathbf{\Sigma}}_j\mathbf{A}) + (\mathbf{A}^T\hat{\mathbf{\Sigma}}_j\mathbf{A})^{-1}(\mathbf{A}^T\hat{\mathbf{\Sigma}}_i\mathbf{A}) - 2I_d) \end{aligned}$$

where $kl_{i,j}$ is short for $KL(p_i(\mathbf{A}^T\mathbf{x}), p_j(\mathbf{A}^T\mathbf{x}))$, and $\hat{\mu}_{i,j} = \hat{\mu}_i - \hat{\mu}_j$ denotes the distance between two centroids in the input space.

Each $kl_{i,j}$ is a non-convex function of $\mathbf{A}$, so the objective function in Equation 3 is non-convex. However, we can still attain the local maximum through gradient based iterative optimization procedure given the initial estimate of $\mathbf{A}$, which is obtained from the conventional LDA procedure. The gradient of $kl_{i,j}$ with respect to $\mathbf{A}$ is given by

$$\nabla_{\mathbf{A}} kl_{i,j} = \nabla_{\mathbf{A}}^{(1)} kl_{i,j} + \nabla_{\mathbf{A}}^{(2)} kl_{i,j} \quad (9)$$

$$\begin{aligned} \nabla_{\mathbf{A}}^{(1)} kl_{i,j} &= 2\hat{\mu}_{i,j}\hat{\mu}_{i,j}^T \mathbf{A}((\mathbf{A}^T\hat{\mathbf{\Sigma}}_i\mathbf{A})^{-1} + (\mathbf{A}^T\hat{\mathbf{\Sigma}}_j\mathbf{A})^{-1}) \quad (10) \\ &- 2\hat{\mathbf{\Sigma}}_i\mathbf{A}(\mathbf{A}^T\hat{\mathbf{\Sigma}}_i\mathbf{A})^{-1}\mathbf{A}^T\hat{\mu}_{i,j}\hat{\mu}_{i,j}^T\mathbf{A}(\mathbf{A}^T\hat{\mathbf{\Sigma}}_i\mathbf{A})^{-1} \\ &- 2\hat{\mathbf{\Sigma}}_j\mathbf{A}(\mathbf{A}^T\hat{\mathbf{\Sigma}}_j\mathbf{A})^{-1}\mathbf{A}^T\hat{\mu}_{i,j}\hat{\mu}_{i,j}^T\mathbf{A}(\mathbf{A}^T\hat{\mathbf{\Sigma}}_j\mathbf{A})^{-1} \\ \nabla_{\mathbf{A}}^{(2)} kl_{i,j} &= 2\hat{\mathbf{\Sigma}}_j\mathbf{A}(\mathbf{A}^T\hat{\mathbf{\Sigma}}_i\mathbf{A})^{-1} + 2\hat{\mathbf{\Sigma}}_i\mathbf{A}(\mathbf{A}^T\hat{\mathbf{\Sigma}}_j\mathbf{A})^{-1} \quad (11) \\ &- 2\hat{\mathbf{\Sigma}}_i\mathbf{A}(\mathbf{A}^T\hat{\mathbf{\Sigma}}_i\mathbf{A})^{-1}\mathbf{A}^T\hat{\mathbf{\Sigma}}_j\mathbf{A}(\mathbf{A}^T\hat{\mathbf{\Sigma}}_i\mathbf{A})^{-1} \\ &- 2\hat{\mathbf{\Sigma}}_j\mathbf{A}(\mathbf{A}^T\hat{\mathbf{\Sigma}}_j\mathbf{A})^{-1}\mathbf{A}^T\hat{\mathbf{\Sigma}}_i\mathbf{A}(\mathbf{A}^T\hat{\mathbf{\Sigma}}_j\mathbf{A})^{-1} \end{aligned}$$

Hence the total gradient of the objective function defined in Equation 3 with respect to $\mathbf{A}$ is given by

$$\nabla_{\mathbf{A}} f(\mathbf{A}) = \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} w_i w_j \nabla_{\mathbf{A}} kl_{i,j} \quad (12)$$

For iterative optimization, we employ the conjugate gradient algorithm to obtain $\mathbf{A}$ given the above gradient.Compared to other iterative optimization procedures, conjugate gradient has faster convergence rate than first order methods and less complexity than second order methods. For the iterative LDA problem we studied, it takes less than 10 steps and a few seconds to converge. The details of conjugate gradient algorithm is beyond the scope of this paper and can be found in [7]. The resulting algorithm is described in Figure 1.

2.3. Kernel KLDA

The generalization to kernel KLDA is quite straightforward by employing the kernel trick. The only difference is that we compute the KL divergence for the transformation of the data embedded in the RKHS.

Input: data $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, t_{max}.

- Perform conventional LDA/KFDA algorithm to obtain an initial transformation matrix $\mathbf{A}_0$. Set $t = 0$.
- Run conjugate gradient algorithm to minimize the objective function defined in Equation 3.
  - Calculate the gradient $\nabla\mathbf{A}_t$ via Equations 6-12. (For kernel KLDA, use Equations 15 – 16 instead of Equations 6 – 7)
  - Set $\mathbf{g}_0 = \nabla\mathbf{A}_0$ and $\mathbf{h}_0 = \mathbf{g}_0$ if $t = 0$.
  - Find λ_t that maximizes $f(\mathbf{A}_t + \lambda_t\mathbf{h}_t)$ along direction $\mathbf{h}_t$.
  - Update $\mathbf{A}_{t+1} = \mathbf{A}_t + \lambda_t\mathbf{h}_t$, $\mathbf{g}_{t+1} = \nabla f(\mathbf{A}_{t+1})$ and $\mathbf{h}_{t+1} = \mathbf{g}_t + \gamma_t\mathbf{h}_t$, where $\gamma_t = \dfrac{\mathbf{g}_{t+1} \cdot \mathbf{g}_{t+1}}{\mathbf{g}_t \cdot \mathbf{g}_t}$.
  - Increment t and repeat the above steps. Quit if $t > t_{max}$, or either $|f(\mathbf{A}_{t+1} - f(\mathbf{A}_t)|$ or $||\mathbf{A}_{t+1} - \mathbf{A}_t||$ is sufficiently small.

Output: transformation matrix $\mathbf{A}_t$.

Fig. 1. Kullback-Leibler Discriminant Analysis

First, the sample mean vector and the covariance matrix for class j in the RKHS is given by

$$\hat{\mu}_j^{'} = \sum_{\mathbf{x}_i \in C_j} \phi(\mathbf{x}_i)/n_i = \Phi(\mathbf{X})\mathbf{e}_i/n_i \quad (13)$$

$$\begin{aligned} \hat{\mathbf{\Sigma}}_j^{'} &= \sum_{\mathbf{x}_i \in C_j} (\phi(\mathbf{x}_i) - \hat{\mu}_j^{'})(\phi(\mathbf{x}_i) - \hat{\mu}_j^{'})^T/n_i \quad (14) \\ &= \sum_{\mathbf{x}_i \in C_j} \Phi(\mathbf{X})J_i\Phi(\mathbf{X})/n_i \\ J_i &= diag(\mathbf{e}_i) - \mathbf{e}_i\mathbf{e}_i^T/n_i \end{aligned}$$

where $\mathbf{e}_i$ is the indicator vector for class i defined earlier, $diag(\mathbf{e}_i)$ denotes the diagonal matrix with $\mathbf{e}_i$ as its diagonal elements. Equation 14 is derived based on the fact that J_i is a projection matrix such that $J_iJ_i^T = J_i$.

Here again, the transformation matrix $\mathbf{W}$ can be represented in terms of a linear combination of $\Phi(\mathbf{X})$ and specified by $\mathbf{W} = \Phi(\mathbf{X})\mathbf{A}$, then we have

$$\mathbf{W}^T\hat{\mathbf{\Sigma}}_i^{'}\mathbf{W} = \mathbf{A}^T\mathbf{K}J_i\mathbf{K}\mathbf{A}$$

Instead of solving $\mathbf{W}$, we treat $\mathbf{A}$ as the new unknown variable to be solved. Next we define

$$\hat{\mu}_{i,j} = \mathbf{e}_i/n_i - \mathbf{e}_j/n_j \quad (15)$$

$$\hat{\mathbf{\Sigma}}_i = \mathbf{K}J_i\mathbf{K} + \lambda\mathbf{I}_d \quad (16)$$

It is now straightforward to see that the linear KLDA algorithm outlined in Figure 1 can be generalized to the kernel algorithm with minimal changes by substituting $\hat{\mu}_{i,j}$ and $\hat{\mathbf{\Sigma}}_{i(j)}$ in Equations 6 - 7 with their new values defined in the above equations.

Table 1. Results for Multispectral Image Classification

		5%	10%	20%
Test Error	LDA	$22.77 \pm 1.26\%$	$18.56 \pm 0.90\%$	$15.92 \pm 0.42\%$
	L-KLDA	$20.55 \pm 1.67\%$	$16.26 \pm 1.05\%$	$12.28 \pm 0.50\%$
p-value		$1.4817e-04$	$8.9547e-05$	$1.3584e-08$
Test Error	KFDA	$16.20 \pm 0.89\%$	$12.45 \pm 0.59\%$	$10.48 \pm 0.36\%$
	K-KLDA	$15.64 \pm 0.80\%$	$11.88 \pm 0.65\%$	$9.65 \pm 0.52\%$
p-value		$7.2504e-04$	$2.6925e-05$	$5.5879e-06$

Table 2. Testing Results for Handwritten Digit Recognition

		5%	10%	20%
Test Error	LDA	$22.24 \pm 1.23\%$	$16.30 \pm 0.89\%$	$13.24 \pm 0.76\%$
	L-KLDA	$18.18 \pm 1.11\%$	$13.91 \pm 0.86\%$	$11.89 \pm 0.49\%$
p-value		$7.9012e-07$	$2.4691e-05$	$1.1266e-04$
Test Error	KFDA	$11.32 \pm 0.45\%$	$8.17 \pm 0.54\%$	$6.55 \pm 0.32\%$
	K-KLDA	$10.47 \pm 0.77\%$	$7.97 \pm 0.38\%$	$6.35 \pm 0.23\%$
p-value		$8.41e-02$	$3.45e-02$	$1.73e-02$

3. EXPERIMENTAL RESULTS

In this section, we demonstrate the performance of our algorithm for classification in two different applications. The first application is supervised image labeling for multispectral images. The second application is handwritten digit recognition. For both applications, we applied both our linear and kernel KLDA algorithm for feature transformation and compared them with the conventional LDA and KFDA.

To make the comparison on an equal basis and statistically meaningful, for each application, we compared for different sample sizes by using 5%, 10% and 20% of all labelled samples for training and the rest for testing. For each sample size, we repeated the test 10 times, each time with a different random partition of the training and testing data. Both LDA and KFDA under comparison are properly regularized to make the scatter matrices well-conditioned, with the regularization and kernel parameters chosen via 5-fold cross validation. The same RBF kernel and kernel parameters were used for both KFDA and kernel KLDA methods. The nearest neighbor classifier was employed for classification on the extracted features from different methods.

In the first experiment, we used a multispectral image of an agricultural area in West Indiana, USA. The image is maintained by the Laboratory for Applications of Remote Sensing, Purdue University and is available for downloading from the web. It was captured by the AVIRIS sensor and each pixel is associated with a reflection spectrum comprised of 220 bands in the range $375 - 2200$nm. This allows us to achieve accurate pixel level classification of different material types based on the spectral information at each pixel location. 9 classes of different terrain types are labeled across the test image containing a total of 9345 labeled samples. We used the reflectance spectra directly as input vectors and applied various discriminant analysis algorithms to them. The test results of different methods over varying training sample sizes are listed in Table 1, including the mean and the standard deviation of testing error. We also applied a paired t-test to the testing errors of the 10 trials for each training sample size, and listed the p-values of the t-test results in Table 1.

The results clearly show that our KLDA algorithms, both the linear and kernel versions, outperformed their counterparts in terms of lower error rates with commensurate stability (this is indicated by their similar standard deviations). All p-values obtained from the t-tests, as listed in Table 1, are much smaller than 5%. This indicates that the alternative hypothesis will be accepted with 95% confidence, that the testing results come from different distributions and their differences are statistically significant, which evidences the superiority of the proposed algorithms over the conventional ones.

In the second experiment, we tested the performance of KLDA for handwritten digit recognition. We used the USPS database of handwritten digits from 0 to 9. The database contains 7291 gray-scale images at a resolution of 16×16. We concatenated the pixels in raster scan order and formed a 256-dimensional input vector for each image. The test results of different methods are listed in Table 2 with similar trend as that in the previous experiment indicating improved performance of our proposed algorithm over the alternatives for both the linear and kernel versions.

4. CONCLUSIONS

In this paper, we present a novel discriminant analysis algorithm for feature extraction in supervised learning. A new objective function is proposed based on the KLdivergences between pairs of classes. The optimal feature transformation can be found by maximizing the objective function. The proposed algorithm can also be kernelized by exploiting the kernel trick. Experiments demonstrate clear improvements of our algorithm over conventional methods.

ACKNOWLEDGMENT

The authors want to thank Zhouyu Fu for pointing out the sources of the data used in the experiments and the owners of the data for making them available for research purposes.

References

[1] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley, 2nd edition, 2000.

[2] M. Loog, R.P.W. Duin, and R. Haeb-Umbach, "Multiclass linear dimension reduction by weighted pairwise fisher criteria," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 23, no. 7, pp. 762–766, 2001.

[3] N. Kumar and A. Andreou, "Heteroscedastic discriminant analysis and reduced rank hmms for improved speech recognition," *Speech Communication*, vol. 26, pp. 283–297, 1998.

[4] M. Loog and R.P.W. Duin, "Linear dimensionality reduction via a heteroscedastic extension of lda: The chernoff criterion," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 732–739, 2004.

[5] S. Kullback and R. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, pp. 79–86, 1951.

[6] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Muller, "Fisher discriminant analysis with kernels," in *In IEEE Neural Networks for Signal Processing Workshop*, 1999, pp. 41–48.

[7] W. H. Press, S. A. Teukolsky, W. T. Vetterlin, and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, 1992.