Efficient Kernel Discriminant Analysis via Spectral Regression*

Deng Cai
UIUC
dengcai2@cs.uiuc.edu

Xiaofei He
Yahoo!
hex@yahoo-inc.com

Jiawei Han
UIUC
hanj@cs.uiuc.edu

Abstract

Linear Discriminant Analysis (LDA) has been a popular method for extracting features which preserve class separability. The projection vectors are commonly obtained by maximizing the between class covariance and simultaneously minimizing the within class covariance. LDA can be performed either in the original input space or in the reproducing kernel Hilbert space (RKHS) into which data points are mapped, which leads to Kernel Discriminant Analysis (KDA). When the data are highly nonlinear distributed, KDA can achieve better performance than LDA. However, computing the projective functions in KDA involves eigen-decomposition of kernel matrix, which is very expensive when a large number of training samples exist. In this paper, we present a new algorithm for kernel discriminant analysis, called Spectral Regression Kernel Discriminant Analysis *(SRKDA). By using spectral graph analysis, SRKDA casts discriminant analysis into a regression framework which facilitates both efficient computation and the use of regularization techniques. Specifically, SRKDA only needs to solve a set of regularized regression problems and there is no eigenvector computation involved, which is a huge save of computational cost. Our computational analysis shows that SRKDA is 27 times faster than the ordinary KDA. Moreover, the new formulation makes it very easy to develop incremental version of the algorithm which can fully utilize the computational results of the existing training samples. Experiments on face recognition demonstrate the effectiveness and efficiency of the proposed algorithm.*

1. Introduction

Linear discriminant analysis (LDA) is a traditional statistical method that has proved successful on classification problems [5]. The projection vectors are commonly obtained by maximizing the between class covariance and simultaneously minimizing the within class covariance. The classical LDA is a linear method and fails for nonlinear problems.

To deal with this limitation, nonlinear extensions of LDA through "kernel trick" have been proposed. The main idea of kernel-based methods is to map the input data to a feature space through a nonlinear mapping, where the inner products in the feature space can be computed by a kernel function without knowing the nonlinear mapping explicitly [10]. Kernel Fisher Discriminant Analysis (KFD) in [8] and Generalized Discriminant Analysis (GDA) in [1] are two independently developed approaches for kernel-based nonlinear extensions of LDA. They are essentially equivalent. To avoid confusion, we will refer this approach as Kernel Discriminant Analysis (KDA) hereafter.

When solving the optimization problem of KDA, we need to handle the possible singularity problem of total scatter matrix. There are many approaches try to address this issue either by using regularization techniques [8] or by applying singular value decomposition [1]. Both these two approaches involve the eigen-decomposition of the kernel matrix which is computationally expensive. Moreover, due to the difficulty of designing an incremental solution for the eigen-decomposition of the kernel matrix, there has been little work on designing incremental KDA algorithms that can efficiently incorporate new data examples as they become available.

In [9], S. Mika et al. made a first attempt to speed up KDA through a greedy approximation technique. However, their algorithm was developed to handle the binary classification problem. For a multi-class problem, the authors suggested the one against the rest scheme by considering all two-class problems. Recently, Xiong *et al*. [14] proposed a new algorithm called KDR/QR, a KDA variation in which QR decomposition is applied rather than eigen-decomposition. However, there is no theoretical relation between the optimization problem solved by KDA/QR and that of KDA. It is not clear under what situation KDA/QR can achieve similar performance as KDA.

*The work was supported in part by the U.S. National Science Foundation NSF IIS-05-13678, NSF BDI-05-15813 and MIAS (a DHS Institute of Discrete Science Center for Multimodal Information Access and Synthesis). Any opinions, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

In this paper, we propose a new algorithm for kernel discriminant analysis, called *Spectral Regression Kernel Discriminant Analysis* (SRKDA). By using spectral graph analysis, SRKDA casts discriminant analysis into a regression framework which facilitates both efficient computation and the use of regularization techniques. Specifically, SRKDA only needs to solve a set of regularized regression problems and there is no eigenvector computation involved, which is a huge save of computational cost. Moreover, the new formulation makes it very easy to develop incremental version of the algorithm which can fully utilize the previous computational results on the existing training samples.

The points below highlight the contributions of this paper:

- For binary classification problem, KDA has been shown to be equivalent to the regularized kernel regression [10]. We extend this relation to multi-class case.
- We provide a new formulation of the KDA optimization problem. With this new formulation, the KDA optimization problem can be efficiently solved by avoiding the eigen-decomposition of the kernel matrix.
- Moreover, LSKDA can be naturally implemented in the incremental fashion which has only quadratic-time complexity. It is a huge improvement comparing to the cubic-time complexity of the ordinary KDA approaches.
- Since LSKDA uses regression as a building block, various kinds of regularization techniques can be easily incorporated (*e.g.*, L_1-norm regularizer to produce sparse projections).

We begin our discussion with a brief review of KDA.

2. A Brief Review of KDA

Suppose we have a set of m samples $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_m \in \mathbb{R}^n$, belonging to c classes. We consider the problem in a feature space $\mathcal{F}$ induced by some nonlinear mapping $\phi : \mathbb{R}^n \rightarrow \mathcal{F}$. For a proper chosen ϕ, an inner product $\langle , \rangle$ can be defined on $\mathcal{F}$ which makes for a so-called reproducing kernel Hilbert space (RKHS). More specifically, $\langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ holds where $\mathcal{K}(.,.)$ is a positive semi-definite kernel function.

Let S_b^ϕ, S_w^ϕ and S_t^ϕ denote the between-class, within-class and total scatter matrices in the feature space respectively. We have

$$S_b^\phi = \sum_{k=1}^{c} m_k (\boldsymbol{\mu}_\phi^{(k)} - \boldsymbol{\mu}_\phi)(\boldsymbol{\mu}_\phi^{(k)} - \boldsymbol{\mu}_\phi)^T,$$

$$S_w^\phi = \sum_{k=1}^{c} \left(\sum_{i=1}^{m_k} \Big(\phi(\mathbf{x}_i^{(k)}) - \boldsymbol{\mu}_\phi^{(k)}\Big)\Big(\phi(\mathbf{x}_i^{(k)}) - \boldsymbol{\mu}_\phi^{(k)}\Big)^T \right),$$

$$S_t^\phi = S_b^\phi + S_w^\phi = \sum_{i=1}^{m} \Big(\phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi\Big)\Big(\phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi\Big)^T,$$

where $\boldsymbol{\mu}_\phi^{(k)}$ and $\boldsymbol{\mu}_\phi$ are the centroid of the k-th class and the global centroid, respectively in the feature space. m_k is the number of samples in the k-th class.

KDA seeks the optimal projective function $\boldsymbol{\nu}$ in the feature space by solving the following optimization problem

$$\boldsymbol{\nu}_{opt} = \arg\max \frac{\boldsymbol{\nu}^T S_b^\phi \boldsymbol{\nu}}{\boldsymbol{\nu}^T S_w^\phi \boldsymbol{\nu}} = \arg\max \frac{\boldsymbol{\nu}^T S_b^\phi \boldsymbol{\nu}}{\boldsymbol{\nu}^T S_t^\phi \boldsymbol{\nu}}, \quad (1)$$

By representer theorem, we know that the solution to the above optimization problem is a linear combination of $\phi(\mathbf{x}_i)$ [1][10]. Thus, there exist coefficients α_i such that $\boldsymbol{\nu}_{opt} = \sum_{i=1}^{m} \alpha_i \phi(\mathbf{x}_i)$. Let $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_m]^T$, it can be proved [1] that Eqn. (1) is equivalent to:

$$\boldsymbol{\alpha}_{opt} = \arg\max \frac{\boldsymbol{\alpha}^T K W K \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T K K \boldsymbol{\alpha}}, \quad (2)$$

and the optimal $\boldsymbol{\alpha}$'s are given by the eigenvectors with respect to the maximum eigenvalues of the following eigen-problem:

$$KWK\boldsymbol{\alpha} = \lambda KK\boldsymbol{\alpha}. \quad (3)$$

where K is the kernel matrix ($K_{ij} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$) and W is defined as:

$$W_{ij} = \begin{cases} 1/m_k, & \text{if } \mathbf{x}_k \text{ and } \mathbf{x}_j \text{ both belong to the } k\text{-th class;} \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

Each eigenvector $\boldsymbol{\alpha}$ gives a projective function $\boldsymbol{\nu}$ in the feature space. For a data example $\mathbf{x}$, we have

$$\langle \boldsymbol{\nu}, \phi(\mathbf{x}) \rangle = \sum_{i=1}^{m} \alpha_i \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle = \sum_{i=1}^{m} \alpha_i \mathcal{K}(\mathbf{x}_i, \mathbf{x}) = \boldsymbol{\alpha}^T K(:, \mathbf{x})$$

where $K(:, \mathbf{x}) \doteq [K(\mathbf{x}_1, \mathbf{x}), \cdots, K(\mathbf{x}_m, \mathbf{x})]^T$.

The eigen-problem in Eqn. (3) can be solved as follows [1]. Suppose the rank of K is $r (r \leq m)$ and the eigen-decomposition of K is $K = U \Sigma U^T$. Without loss of generality, let Σ be a diagonal matrix of nonzero eigenvalues and U is the matrix of normalized eigenvectors associated with Σ. Thus Σ^{-1} exists and $U^T U = I$, where I is the identity matrix.

Let $\boldsymbol{\beta} = \Sigma U^T \boldsymbol{\alpha}$ and substituting K in Eqn. (2), we get

$$\frac{\big(\Sigma U^T \boldsymbol{\alpha}\big)^T U^T W U \big(\Sigma U^T \boldsymbol{\alpha}\big)}{\big(\Sigma U^T \boldsymbol{\alpha}\big)^T U^T U \big(\Sigma_r U^T \boldsymbol{\alpha}\big)} = \frac{\boldsymbol{\beta}^T U^T W U \boldsymbol{\beta}}{\boldsymbol{\beta}^T \boldsymbol{\beta}}.$$

Thus, the optimal $\boldsymbol{\beta}$'s are the leading eigenvectors of matrix $U^T W U$. Once $\boldsymbol{\beta}$'s are calculated, $\boldsymbol{\alpha}$ can be computed as $\boldsymbol{\alpha} = U \Sigma^{-1} \boldsymbol{\beta}$.

In the following ,we use the term *flam* [11], a compound operation consisting of one addition and one multiplication, to measure the operation counts. All the kernel methods need to compute the kernel matrix K which requires $O(m^2n)$ flam, where n is the number of features. The eigen-decomposition of K requires $\frac{9}{2}m^3$ flam [12]; With the block-diagonal structure of W, the leading $c-1$ eigenvectors of U^TWU can be computed within $\frac{9}{2}c^3 + \frac{3}{2}mc^2$ flam; Computing $\boldsymbol{\alpha}$'s from $\boldsymbol{\beta}$'s requires m^2c flam. Considering $m \gg c$, we conclude the time complexity of KDA measured by flam is

$$\frac{9}{2}m^3 + m^2c + O(m^2n). \tag{5}$$

For a large scale problem, we have $m \gg n$. Thus, the time complexity of KDA is determined by $\frac{9}{2}m^3$, which is the cost of eigen-decomposition of size $m \times m$ kernel matrix K.

3. Efficient KDA via Spectral Regression

In order to solve the KDA eigen-problem in Eqn. (3) efficiently, we use the following theorem:

Theorem 1 *Let $\mathbf{y}$ be the eigenvector of eigen-problem*

$$W\mathbf{y} = \lambda\mathbf{y} \tag{6}$$

with eigenvalue λ. If $K\boldsymbol{\alpha} = \mathbf{y}$, then $\boldsymbol{\alpha}$ is the eigenvector of eigen-problem in Eqn. (3) with the same eigenvalue λ.

Proof We have $W\mathbf{y} = \lambda\mathbf{y}$. At the left side of Eqn. (3), replace $K\boldsymbol{\alpha}$ by $\mathbf{y}$, we have

$$KWK\boldsymbol{\alpha} = KW\mathbf{y} = K\lambda\mathbf{y} = \lambda K\mathbf{y} = \lambda KK\boldsymbol{\alpha}$$

Thus, $\boldsymbol{\alpha}$ is the eigenvector of eigen-problem Eqn. (3) with the same eigenvalue λ. ∎

Theorem 1 shows that instead of solving the eigen-problem Eqn. (3), the KDA projective functions can be obtained through two steps:

1. Solve the eigen-problem in Eqn. (6) to get $\mathbf{y}$.
2. Find $\boldsymbol{\alpha}$ which satisfies $K\boldsymbol{\alpha} = \mathbf{y}$. The kernel matrix K is positive semi-definite. When K is non-singular (positive definite), for any given $\mathbf{y}$, we have a unique $\boldsymbol{\alpha} = K^{-1}\mathbf{y}$ which satisfies the above linear equations system. When K is singular, the system may have no solution or have infinite many solutions (the linear equations system is underdetermined) [6]. A possible way is to approximate $\boldsymbol{\alpha}$ by solving the following linear equations:
$$(K + \delta I)\boldsymbol{\alpha} = \mathbf{y} \tag{7}$$
where I is the identity matrix and $\delta \geq 0$ is the regularization parameter.

The advantages of this two-step approach are as follows:

1. We will show later how the eigen-problem in Eqn. (6) is *trivial* and we can directly get those eigenvectors $\mathbf{y}$.
2. The eigen-decomposition of K is avoided. Since the matrix $K + \delta I$ is positive definite, the Cholesky decomposition can be used to efficiently solve the linear equations in Eqn. (7) [6], [11]. The computational complexity analysis will be provided in the later section.

The linear equations system in Eqn. (7) has close connection with regularized regression [13]. We denote the projective function in the feature space as:

$$f(\mathbf{x}) = \langle \boldsymbol{\nu}, \phi(\mathbf{x}) \rangle = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}, \mathbf{x}_i)$$

It can be easily verified that the solution $\boldsymbol{\alpha}^* = (K+\delta I)^{-1}\mathbf{y}$ given by equations in Eqn. (7) is the optimal solution of the following regularized regression problem [13]:

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} \big(f(\mathbf{x}_i) - y_i\big)^2 + \delta \|f\|_K^2 \tag{8}$$

where y_i is the i-th element of $\mathbf{y}$, $\mathcal{F}$ is the RKHS associated with Mercer kernel $\mathcal{K}$ and $\| \ \|_K$ is the corresponding norm.

Now let us analyze the eigenvectors of W which is defined in Eqn. (4). The W is block-diagonal, thus, its eigenvalues and eigenvectors are the union of the eigenvalues and eigenvectors of its blocks (the latter padded appropriately with zeros). It is straightforward to show that there are exactly c eigenvectors of W with the same eigenvalue 1 [2]. These eigenvectors are

$$\mathbf{y}_k = [\underbrace{0, \cdots, 0}_{\sum_{i=1}^{k-1} m_i}, \underbrace{1, \cdots, 1}_{m_k}, \underbrace{0, \cdots, 0}_{\sum_{i=k+1}^{c} m_i}]^T \quad k = 1, \cdots, c \tag{9}$$

Since 1 is a repeated eigenvalue of W, we could just pick any other c orthogonal vectors in the space spanned by $\{\mathbf{y}_k\}$, and define them to be our c eigenvectors. The vector of all ones $\mathbf{e}$ is naturally in the spanned space. This vector is useless since the corresponding projective function will embed all the samples to the same point. Therefor, we pick $\mathbf{e}$ as our first eigenvector of W and use Gram-Schmidt process to orthogonalize the remaining eigenvectors. The vector $\mathbf{e}$ can then be removed, which leaves us exactly $c-1$ eigenvectors of W. We denote them as:

$$\{\bar{\mathbf{y}}_k\}_{k=1}^{c-1}, \ (\bar{\mathbf{y}}_k^T\mathbf{e} = 0, \ \ \bar{\mathbf{y}}_i^T\bar{\mathbf{y}}_j = 0, \ i \neq j) \tag{10}$$

The above two-step approach essentially combines the spectral analysis of the matrix W and regression techniques. Therefor, we named this new approach as *Spectral Regression Kernel Discriminant Analysis* (SRKDA). It is important to note that the similar technique can also be applied on Linear Discriminant Analysis [3].

3.1. Theoretical Analysis

SRKDA calculates the projective functions through the linear equations system in Eqn. (7). When the kernel matrix K is positive definite and the $\delta = 0$, Theorem 1 shows that the $c-1$ solutions $\boldsymbol{\alpha}_k = K^{-1}\mathbf{y}_k$ are exactly the eigenvectors of the KDA eign-problem in Eqn. (3) with respect to the eigenvalue 1. In this case, SRKDA is equivalent to ordinary KDA.

One of the most popular kernels is the Gaussian RBF kernel, $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$. Regarding the Gaussian kernel, we have the following theorem:

Theorem 2 *If all the sample vectors are different and the Gaussian RBF kernel is used, all $c-1$ projective functions in SRKDA are eigenvectors of eigen-problem in Eqn. (3) with respect to eigenvalue 1 when $\delta = 0$. In other words, the SRKDA and ordinary KDA are equivalent.*

Proof When all the sample vectors are different, the kernel matrix K of the Gaussian RBF kernel has full rank [7][10]. By theorem 1, the $c-1$ solutions $\boldsymbol{\alpha}_k = K^{-1}\mathbf{y}_k$ are exactly the eigenvectors of the KDA eign-problem in Eqn. (3) with respect to the eigenvalue 1. ∎

3.2. Computational Analysis

The computation of SRKDA involves two steps: responses ($\bar{\mathbf{y}}_k$ in Eqn. 10) generation and regularized regression. The cost of the first step is mainly the cost of Gram-Schmidt method, which requires $(mc^2 - \frac{1}{3}c^3)$ flam [11].

To solve the $c-1$ linear equations systems in Eqn. (7), we can use the Cholesky decomposition, which uniquely factorizes the positive definite matrix $K + \delta I$ in the form $K + \delta I = R^T R$, where R is upper triangular with positive diagonal elements. The Cholesky decomposition requires $\frac{1}{6}m^3$ flam [11]. With this Cholesky decomposition, the $c-1$ linear equations can be solved within $m^2 c$ flam [11]. Besides solving the SRKDA optimization problem, we also need to compute the kernel matrix K which requires $O(m^2 n)$ flam, where n is the number of features. Thus, the computational cost of SRKDA is

$$\frac{1}{6}m^3 + m^2 c + O(m^2 n) + mc^2 - \frac{1}{3}c^3,$$

which can be simplified as

$$\frac{1}{6}m^3 + m^2 c + O(m^2 n).$$

Comparing to the computational cost of ordinary KDA in Eqn. (5), SRKDA reduces the dominant part, which is $\frac{9}{2}m^3$ of ordinary KDA, to $\frac{1}{6}m^3$; achieves a 27 times speedup.

4. Incremental KDA via Spectral Regression

Due to the difficulty of designing an incremental solution for the eigen-decomposition on the kernel matrix in KDA, there has been little work on designing incremental KDA algorithms that can efficiently incorporate new data examples as they become available. The SRKDA algorithm uses regression instead of eigen-decomposition to solve the optimization problem, which enables us to develop incremental version of SRKDA.

The major cost in SRKDA computation is the step of Cholesky decomposition which requires $\frac{1}{6}m^3$ flam. Fortunately, the Cholesky decomposition can be easily implemented in the incremental manner [11]. Actually, *Sherman's march*, one of the most popular Cholesky decomposition algorithms, is implemented in the incremental manner [11]. We show how to proceed from $(m-1)\times(m-1)$ matrix to a $m \times m$ matrix. We have

$$\begin{aligned} K_m &= \begin{pmatrix} K_{m-1} & \mathbf{k}_{1m} \\ \mathbf{k}_{1m}^T & k_{mm} \end{pmatrix} \\ &= \begin{pmatrix} R_{m-1}^T & \mathbf{0} \\ \mathbf{r}_{1m}^T & r_{mm} \end{pmatrix} \begin{pmatrix} R_{m-1} & \mathbf{r}_{1m} \\ \mathbf{0} & r_{mm} \end{pmatrix}, \end{aligned}$$

which leads to

$$\begin{aligned} K_{m-1} &= R_{m-1}^T R_{m-1} \\ \mathbf{k}_{1m} &= R_{m-1}^T \mathbf{r}_{1m} \\ k_{mm} &= r_{mm}^2 \end{aligned}$$

When the Cholesky decomposition of the $(m-1)\times(m-1)$ submatrix K_{m-1} is known, it is easy to get the Cholesky decomposition of the $m \times m$ K_m. For detailed derivation, please see [11].

Now, let us consider the additional computational cost of incremental SRKDA when Δm new data samples are injected to the system which already has m samples. Compare to the batch mode of SRKDA, we can get computational saving on two steps:

1. We only need to calculate the additional part of kernel matrix which requires $O(nm\Delta m + n\Delta m^2)$ flam;
2. The incremental Cholesky decomposition requires $\frac{1}{6}(m+\Delta m)^3 - \frac{1}{6}m^3$ flam [11].

Thus, the computation cost of incremental SRKDA measured by flam is

$$\begin{aligned} &\frac{1}{2}m^2\Delta m + \frac{1}{2}m\Delta m^2 + \frac{1}{6}\Delta m^3 + (m+\Delta m)^2 c \\ &+ O(nm\Delta m + n\Delta m^2) + (m+\Delta m)c^2 - \frac{1}{3}c^3. \end{aligned}$$

When $\Delta m \ll m$ and $c \ll m$, the above cost can be simplified as

$$(\frac{\Delta m}{2} + c)m^2 + O(nm\Delta m).$$

Table 1. Computational complexity of KDA and SRKDA

Algorithm		operation counts (*flam* [11])
Batch mode	KDA	$\frac{9}{2}m^3 + cm^2 + O(nm^2)$
	SRKDA	$\frac{1}{6}m^3 + cm^2 + O(nm^2)$
Incremental mode	KDA	$\frac{9}{2}m^3 + cm^2 + O(nm\Delta m)$
	SRKDA	$(\frac{\Delta m}{2} + c)m^2 + O(nm\Delta m)$

m: the number of data samples
n: the number of features
c: the number of classes
Δm: the number of new data samples

We summarize our complexity analysis results in Table 1. The main conclusions include:

- The ordinary KDA needs to perform eigen-decomposition on the kernel matrix, which is very computationally expensive. Moreover, it is difficult to develop incremental algorithm based on the ordinary KDA formulation. In both batch and incremental modes, ordinary KDA has the dominant part of the cost as $\frac{9}{2}m^3$.
- SRKDA performs regression instead of eigen-decomposition. In the batch mode, it only has the dominant part of the cost as $\frac{1}{6}m^3$, which is a 27 times speedup of ordinary KDA. Moreover, it is easy to develop incremental version of SRKDA which only has quadratic-time complexity with respect to m. This computational advantage makes SRKDA much more practical in real world applications.

5. Experimental Results

In this section, we investigate the performance of our proposed SRKDA algorithm. All of our experiments have been performed on a P4 3.20GHz Windows XP machine with 2GB memory.

Due to the space limitation, we only report the experimental result on the face recognition task. Please see our technical report [2] for more experimental results. The CMU PIE face database[1] is used in our experiment. It contains 68 subjects with 41,368 face images as a whole. The face images were captured under varying pose, illumination and expression. In our experiment, the five near frontal poses (C05, C07, C09, C27, C29) under different illuminations and expressions are used which leaves us 11,554 face images. All the images are manually aligned and cropped. The cropped images are 32×32 pixels, with 256 gray levels per pixel[2]. Among the 11,554 images, 8,000 images are used as the training set and the remaining 3,554 images are used for testing. We also run several cases by training all the algorithms on the first 2000, 3000, $\cdots$, 8000 images in the training set.

[1] http://www.ri.cmu.edu/projects/project_418.html
[2] http://ews.uiuc.edu/~dengcai2/Data/data.html

5.1. Compared algorithms

Five algorithms which are compared in our experiments are listed below:

1. Linear Discriminant Analysis (LDA) [5], which provides us a baseline performance of linear algorithms.
2. Kernel Discriminant Analysis (KDA) as discussed in Section 2. We test the regularized version and choose the regularization parameter δ by five fold cross-validation on the training set.
3. Spectral Regression Kernel Discriminant Analysis (SRKDA), our approach proposed in this paper. The regularization parameter δ is also chosen by five fold cross-validation on the training set.
4. KDA/QR [14], a KDA variation in which QR decomposition is applied rather than eigen-decomposition.
5. Support Vector Machine (SVM) [13], which is believed as one of the state-of-the-art classification algorithms. Specifically, we use the LibSVM system [4] which implemented the multi-class classification with one versus one strategy.

We use the Gaussian RBF kernel for all the kernel-based methods. We tune the kernel width parameter σ and large margin parameter C in SVM to achieve best testing performance for SVM. Then, the same kernel width parameter σ is used in all the other kernel-based algorithms.

5.2. Results

The classification error rate as well as the training time (second) for each method on the three data sets are reported on the Table (2).

The main observations from the performance comparisons include:

- The Kernel Discriminant Analysis model is very effective in classification. SRKDA has the best performance for almost all the cases (even better than SVM).
- Since the eigen-decomposition of the kernel matrix is involved, the ordinary KDA is computationally expensive in training. SRKDA uses regression instead of eigen-decomposition to solve the optimization problem, and thus achieve significant speedup comparing to ordinary KDA. The empirical results are consistent with the theoretical estimation of the efficiency.

Table 2. Performance comparisons on PIE dataset

	Error (%)					Time (s)					Speedup
Training Size	LDA	KDA	SRKDA	KDA/QR	SVM	LDA	KDA	SRKDA	KDA/QR	SVM	
2000	5.29	5.18	**4.81**	15.62	6.30	8.77	36.51	2.47	1.66	24.13	14.8
3000	4.61	4.25	**3.94**	9.82	4.70	9.06	116.9	5.39	3.66	43.99	21.7
4000	4.14	5.53	**3.24**	7.93	3.74	9.42	256.6	10.35	6.39	68.43	24.8
5000	3.85	3.23	**2.90**	5.94	3.29	9.73	502.3	17.40	10.00	96.26	28.9
6000	3.57	2.91	**2.53**	5.68	2.84	10.06	830.7	27.21	14.20	125.6	30.5
7000	3.40	2.65	**2.19**	4.08	2.64	10.39	1340.9	38.65	19.12	155.6	34.7
8000	3.35	2.41	**2.17**	4.00	2.34	10.79	1908.1	53.75	24.96	186.7	35.5

*Column labeled "Speedup" shows how many times faster the SRKDA is (comparing to ordinary KDA).

- The KDA/QR is very efficient because they only need to perform QR decomposition on matrices with size $m \times c$ [14]. However, there is no theoretical relation between the optimization problem solved in KDA/QR and that of the KDA. In all the cases, the performances of KDA/QR is the worst.

6. Conclusions

In this paper, we propose a novel algorithm for kernel discriminant analysis, called *Spectral Regression Kernel Discriminant Analysis* (SRKDA). Our algorithm is developed from a graph embedding viewpoint of KDA problem. It combines the spectral graph analysis and regression to provide an efficient approach for kernel discriminant analysis. Specifically, SRKDA only needs to solve a set of regularized regression problems and there is no eigenvector computation involved, which is a huge save of computational cost. The theoretical analysis shows that SRKDA can achieve 27 times speedup over the ordinary KDA. Moreover, the new formulation makes it very easy to develop incremental version of the algorithm which can fully utilize the computational results of the existing training samples. With incremental implementation, the computational cost of SRKDA reduces to quadratic-time complexity. Extensive experimental results show that our method consistently outperforms the other state-of-the-art KDA extensions considering both effectiveness and efficiency.

References

[1] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, (12):2385–2404, 2000.

[2] D. Cai, X. He, and J. Han. Efficient kernel discriminant analysis via spectral regression. Technical report, Computer Science Department, UIUC, UIUCDCS-R-2007-2888, August 2007.

[3] D. Cai, X. He, and J. Han. SRDA: An efficient algorithm for large scale discriminant analysis. Technical report, Computer Science Department, UIUC, UIUCDCS-R-2007-2857, May 2007.

[4] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[5] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 2nd edition, 1990.

[6] G. H. Golub and C. F. V. Loan. *Matrix computations*. Johns Hopkins University Press, 3rd edition, 1996.

[7] C. A. Micchelli. Algebraic aspects of interpolation. In *Proceedings of Symposia in Applied Mathematics*, volume 36, pages 81–102, 1986.

[8] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In *Proc. of IEEE Neural Networks for Signal Processing Workshop (NNSP)*, 1999.

[9] S. Mika, A. Smola, and B. Schölkopf. An improved training algorithm for kernel fisher discriminants. In *Proceedings AISTATS 2001*. Morgan Kaufmann, 2001.

[10] B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, 2002.

[11] G. W. Stewart. *Matrix Algorithms Volume I: Basic Decompositions*. SIAM, 1998.

[12] G. W. Stewart. *Matrix Algorithms Volume II: Eigensystems*. SIAM, 2001.

[13] V. N. Vapnik. *Statistical learning theory*. John Wiley & Sons, 1998.

[14] T. Xiong, J. Ye, Q. Li, V. Cherkassky, and R. Janardan. Efficient kernel discriminant analysis via QR decomposition. In *Advances in Neural Information Processing Systems 17*, 2004.