Report No. UIUCDCS-R-2007-2888 **UILU-ENG-2007-1790**

Efficient Kernel Discriminant Analysis via Spectral Regression

by

Deng Cai, Xiaofei He, and Jiawei Han

August 2007

Efficient Kernel Discriminant Analysis via Spectral Regression[*]

Deng Cai[†] Xiaofei He[‡] Jiawei Han[†]

[†] Department of Computer Science, University of Illinois at Urbana-Champaign

[‡] Yahoo! Corp.

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. 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. Extensive experiments on spoken letter, handwritten digit image and face image data 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 [7]. 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.

[*]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.

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 [14]. Kernel Fisher Discriminant Analysis (KFD) in [12] 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 Discrimiant Analysis (KDA) hereafter.

When solving the optimization problem of KDA, we need to handle the possible singularity problem of total scatter matrix. There are two approaches try to address this issue either by using regularization techniques [12] or by applying singular value decomposition [1]. Both of these two approaches for solving optimization problem of KDA 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 on 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 [11, 13], 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.* [18] 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.

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:

- KDA in the binary-class case has been shown to be equivalent to regularized kernel regression with the class label as the output [14]. Our paper extends this relation to multi-class case.
- We provides a new formulation of KDA optimization problem. With this new formulation, the KDA optimization problem can be efficiently solved by avoiding the eigen-decomposition of the kernel matrix. Theoretical analysis shows that the new approach can achieve 27-times speedup over the ordinary KDA approaches.
- Moreover, SRKDA can be naturally performed in the incremental manner. The computational results on the existing training samples can be fully utilized when new training samples are injected into the system. Theoretical analysis shows that SRKDA in the incremental mode has only

quadratic-time complexity, which is a huge improvement comparing to the cubic-time complexity of the ordinary KDA approaches.

- Since SRKDA 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). Our approach provides a huge possibility to develop new variations of kernel discriminant analysis.

The remainder of the paper is organized as follows. In Section 2, we provide a brief review of LDA and KDA, plus a detailed computational analysis of KDA. Section 3 introduces our proposed *Spectral Regression Kernel Discriminant Analysis* algorithm. The incremental version of SRKDA is introduced in Section 4 and the extensive experimental results are presented in Section 5. Finally, we provide some concluding remarks in Section 6.

2 A Brief Review of LDA and KDA

Linear Discriminant Analysis (LDA) seeks directions on which the data points of different classes are far from each other while requiring data points of the same class to be close to each other [7]. 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. The objective function of LDA is as follows:

$$\mathbf{a}_{opt} = \arg\max \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_w \mathbf{a}}, \tag{1}$$

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

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

where $\boldsymbol{\mu}$ is the global centroid, m_k is the number of samples in the k-th class, $\boldsymbol{\mu}^{(k)}$ is the centroid of the k-th class, and $\mathbf{x}_i^{(k)}$ is the i-th sample in the k-th class. We call S_w the within-class scatter matrix and S_b the between-class scatter matrix.

Define the total scatter matrix $S_t = \sum_{i=1}^{m} (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^T$, we have $S_t = S_b + S_w$ [7]. The objective function of LDA in Eqn. (1) is equivalent to

$$\mathbf{a}_{opt} = \arg\max \frac{\mathbf{a}^T S_b \mathbf{a}}{\mathbf{a}^T S_t \mathbf{a}}. \tag{2}$$

The optimal $\mathbf{a}$'s are the eigenvectors corresponding to the non-zero eigenvalue of eigen-problem:

$$S_b \mathbf{a} = \lambda S_t \mathbf{a}. \tag{3}$$

Since the rank of S_b is bounded by $c-1$, there are at most $c-1$ eigenvectors corresponding to non-zero eigenvalues [7].

To extend LDA to the nonlinear case, 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,

$$\langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = \mathcal{K}(\mathbf{x}, \mathbf{y})$$

holds where $\mathcal{K}(.,.)$ is a positive semi-definite kernel function. Several popular kernel functions are: Gaussian kernel $\mathcal{K}(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2)$; polynomial kernel $\mathcal{K}(\mathbf{x}, \mathbf{y}) = (1+\mathbf{x}^T\mathbf{y})^d$; Sigmoid kernel $\mathcal{K}(\mathbf{x}, \mathbf{y}) = tanh(\mathbf{x}^T\mathbf{y}+\alpha)$.

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

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

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

$$S_t^\phi = \sum_{i=1}^{m} \left(\phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi\right)\left(\phi(\mathbf{x}_i) - \boldsymbol{\mu}_\phi\right)^T,$$

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.

Let $\boldsymbol{\nu}$ denote the projective function in the feature space, the corresponding objective function (2) in the feature space is

$$\boldsymbol{\nu}_{opt} = \arg\max \frac{\boldsymbol{\nu}^T S_b^\phi \boldsymbol{\nu}}{\boldsymbol{\nu}^T S_t^\phi \boldsymbol{\nu}}, \tag{4}$$

which can be solved by the eigen-problem:

$$S_b^\phi \boldsymbol{\nu} = \lambda S_t^\phi \boldsymbol{\nu}.$$

Because the eigenvectors are linear combinations of $\phi(\mathbf{x}_i)$ [1][14], there exist coefficients α_i such that

$$\boldsymbol{\nu} = \sum_{i=1}^{m} \alpha_i \phi(\mathbf{x}_i).$$

Let $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_m]^T$, it can be proved [1] that Eqn. (4) is equivalent to:

$$\boldsymbol{\alpha}_{opt} = \arg\max \frac{\boldsymbol{\alpha}^T KWK \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T KK \boldsymbol{\alpha}}, \tag{5}$$

and the corresponding eigen-problem is:

$$KWK\boldsymbol{\alpha} = \lambda KK\boldsymbol{\alpha}. \tag{6}$$

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

$$W_{ij} = \begin{cases} 1/m_k, & \text{if } \mathbf{x}_k \text{ and } \mathbf{x}_j \text{ both belong to} \\ & \text{the } k\text{-th class;} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

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

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

where $K(:, \mathbf{x}) \doteq [K(\mathbf{x}_1, \mathbf{x}), \cdots, K(\mathbf{x}_m, \mathbf{x})]^T$. Let $\{\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_{c-1}\}$ be the $c-1$ eigenvectors of the eigen-problem in Eqn. (6) with respect to the non-zero eigenvalues. The transformation matrix $\Theta = [\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_{c-1}]$ is a $m \times (c-1)$ matrix and a data sample $\mathbf{x}$ can be embedded into $c-1$ dimensional subspace by

$$\mathbf{x} \rightarrow \mathbf{z} = \Theta^T K(:, \mathbf{x}).$$

The above approach extends LDA into RKHS by using "kernel trick" is independently developed by Mika *et al.* [12] and Baudat *et al.* [1]. This algorithm was named as Kernel Fisher Discrimiant (KFD) in [12] and Generalized Discriminant Analysis (GDA) in [1].

2.1 Computational Analysis of KDA

To get a stable solution of the eigen-problem in Eqn. (6), the matrix KK is required to be non-singular [8]. When K is singular, there are two methods to solve this problem. The first method is by using eigen-decomposition of K, which was proposed in [1].

Suppose the rank of K is $r (r \leq m)$ and the eigen-decomposition of K is as follows:

$$K = U \Sigma U^T = U_r \Sigma_r U_r^T$$

where $\Sigma = diag(\sigma_1, \cdots, \sigma_m)$ is the diagonal matrix of sorted eigenvalues ($\sigma_1 \geq \cdots \geq \sigma_m \geq 0$) and U is the matrix of normalized eigenvectors associated to Σ. $\Sigma_r = diag(\sigma_1, \cdots, \sigma_r)$ is the diagonal matrix of nonzero eigenvalues and U_r is the first r columns of U. Thus Σ_r^{-1} exists and $U_r^T U_r = I$, where I is the identity matrix.

Substituting K in Eqn. (5), we get

$$\boldsymbol{\alpha}_{opt} = \arg\max \frac{\left(\Sigma_r U_r^T \boldsymbol{\alpha}\right)^T U_r^T W U_r \left(\Sigma_r U_r^T \boldsymbol{\alpha}\right)}{\left(\Sigma_r U_r^T \boldsymbol{\alpha}\right)^T U_r^T U_r \left(\Sigma_r U_r^T \boldsymbol{\alpha}\right)}.$$

We proceed to variable modification using $\boldsymbol{\beta} = \Sigma_r U_r^T \boldsymbol{\alpha}$ and get:

$$\boldsymbol{\beta}_{opt} = \arg\max \frac{\boldsymbol{\beta}^T U_r^T W U_r \boldsymbol{\beta}}{\boldsymbol{\beta}^T \boldsymbol{\beta}},$$

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

The second method is using the idea of regularization, by adding constant values to the diagonal elements of KK, as $KK+\gamma I$, for $\gamma > 0$. It is easy to see that $KK+\gamma I$ is nonsingular. This method is used in [12]. By noticing that

$$KK+\gamma I = U\Sigma U^T U\Sigma U^T + \gamma I = U(\Sigma^2+\gamma I)U^T,$$

we define $\widetilde{\Sigma} = (\Sigma^2+\gamma I)^{1/2}$, the objective function of regularized KDA can be written as:

$$\begin{aligned} &\max \frac{\boldsymbol{\alpha}^T KWK\boldsymbol{\alpha}}{\boldsymbol{\alpha}^T(KK+\gamma I)\boldsymbol{\alpha}} \\ =&\max \frac{\boldsymbol{\alpha}^T U\Sigma U^T W U\Sigma U^T\boldsymbol{\alpha}}{\boldsymbol{\alpha}^T U\widetilde{\Sigma}\widetilde{\Sigma}U^T\boldsymbol{\alpha}} \\ =&\max \frac{\boldsymbol{\beta}^T \widetilde{\Sigma}^{-1}\Sigma U^T W U\Sigma\widetilde{\Sigma}^{-1}\boldsymbol{\beta}}{\boldsymbol{\beta}^T\boldsymbol{\beta}} \end{aligned}$$

where $\boldsymbol{\beta} = \widetilde{\Sigma}U^T\boldsymbol{\alpha}$. The optimal $\boldsymbol{\beta}$'s are the leading eigenvectors of matrix $\widetilde{\Sigma}^{-1}\Sigma U^T W U\Sigma\widetilde{\Sigma}^{-1}$. With this formulation, the above two methods can be computed in exactly the same way.

To reduce the computation in calculating $\boldsymbol{\beta}$, we shall exploit the special structure of W. Without loss of generality, we assume that the data points are ordered according to their labels. It is easy to check that the matrix W has a block-diagonal structure

$$W = \begin{bmatrix} W^{(1)} & 0 & \cdots & 0 \\ 0 & W^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & W^{(c)} \end{bmatrix} \tag{8}$$

where $\{W^{(k)}\}_{k=1}^{c}$ is a $m_k \times m_k$ matrix with all the elements equal to $1/m_k$.

We partition the $m \times r$ matrix U_r as $[U_r^{(1)}, \cdots, U_r^{(c)}]^T$, where $U_r^{(k)} \in \mathbb{R}^{r\times m_k}$. Let $\mathbf{v}_i^{(k)}$ be the i-th column vector of $U_r^{(k)}$, we have:

$$\begin{aligned} U_r^T W U_r =& \sum_{k=1}^{c} U_r^{(k)} W^{(k)} (U_r^{(k)})^T \\ =& \sum_{k=1}^{c} \frac{1}{m_k}\left(\sum_{i=1}^{m_k}\mathbf{v}_i^{(k)} \sum_{i=1}^{m_k}(\mathbf{v}_i^{(k)})^T\right) \\ =& \sum_{k=1}^{c} m_k \bar{\mathbf{v}}^{(k)}(\bar{\mathbf{v}}^{(k)})^T \\ =& HH^T \end{aligned}$$

where $H = \left[\sqrt{m_1}\bar{\mathbf{v}}^{(1)}, \cdots, \sqrt{m_c}\bar{\mathbf{v}}^{(c)}\right] \in \mathbb{R}^{r\times c}$ and $\bar{\mathbf{v}}^{(k)}$ is the average vector of $\mathbf{v}_i^{(k)}$.

To calculate the c leading eigenvectors of HH^T, it is not necessary to work on matrix HH^T which is of size $r\times r$. We can use a much more efficient algorithm. Suppose the Singular Value Decomposition of H is

$$H = P\Gamma Q^T,$$

it is easy to check that the column vectors of P are the eigenvectors of HH^T and the column vectors of Q are the eigenvectors of H^TH [16]. Moreover, if P or Q is given, we can recover the other via the formula $HQ = P\Gamma$ and $P^TH = \Gamma Q^T$. Since $c \ll r$, we can calculate the c eigenvectors of H^TH and then recover the eigenvectors of HH^T, which are $\boldsymbol{\beta}$'s.

We use the term *flam* [15], 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 [16, 8]; Calculating the $c-1$ eigenvectors $\boldsymbol{\beta}$'s requires $\frac{9}{2}c^3 + \frac{3}{2}mc^2$ flam; Computing $\boldsymbol{\alpha}$'s from $\boldsymbol{\beta}$'s requires m^2c flam. Finally, we conclude the time complexity of KDA measured by flam is

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

Considering $m \gg c$, the above time complexity can be simplified as

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

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. (6) efficiently, we use the following theorem:

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

$$W\boldsymbol{y} = \lambda\boldsymbol{y} \tag{10}$$

with eigenvalue λ. If $K\boldsymbol{\alpha} = \boldsymbol{y}$, then $\boldsymbol{\alpha}$ is the eigenvector of eigen-problem in Eqn. (6) with the same eigenvalue λ.

Proof We have $W\mathbf{y} = \lambda\mathbf{y}$. At the left side of Eqn. (6), 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. (6) with the same eigenvalue λ.

Theorem 1 shows that instead of solving the eigen-problem Eqn. (6), the KDA projective functions can be obtained through two steps:

1. Solve the eigen-problem in Eqn. (10) 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 satisfy 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) [8]. A possible way is to approximate $\boldsymbol{\alpha}$ by solving the following linear equations:

$$(K + \delta I)\boldsymbol{\alpha} = \mathbf{y} \tag{11}$$

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. (10) 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. (11) [8], [15]. The computational complexity analysis will be provided in the later section.

The linear equations system in Eqn. (11) has close connection with regularized regression [17]. 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. (11) is the optimal solution of the following regularized regression problem [17]:

$$\min_{f \in \mathcal{F}} \sum_{i=1}^{m} \big(f(\mathbf{x}_i) - y_i\big)^2 + \delta \|f\|_K^2 \tag{12}$$

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. (7) and (8). 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 $W^{(k)}$ has eigenvector $\mathbf{e}^{(k)} \in \mathbb{R}^{m_k}$ associated with eigenvalue 1, where $\mathbf{e}^{(k)} = [1, 1, \cdots, 1]^T$. Also there is only one non-zero eigenvalue of $W^{(k)}$ because the rank of $W^{(k)}$ is 1. Thus, there are exactly c eigenvectors of W with the same eigenvalue 1. These eigenvectors are

$$\mathbf{y}_k = [\ \underbrace{0, \cdots, 0}_{\sum_{i=1}^{k-1} m_i}, \underbrace{1, \cdots, 1}_{m_k}, \underbrace{0, \cdots, 0}_{\sum_{i=k+1}^{c} m_i}\ ]^T \quad k = 1, \cdots, c \tag{13}$$

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:

$$\{\bar{\mathbf{y}}_k\}_{k=1}^{c-1}, \ (\bar{\mathbf{y}}_k^T\mathbf{e} = 0, \ \ \bar{\mathbf{y}}_i^T\bar{\mathbf{y}}_j = 0, \ i \neq j) \tag{14}$$

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). In the following several subsections, we will provide the theoretical and computational analysis on SRKDA. Please see [3] for applying the similar technique on Linear Discriminant Analysis to obtain an efficient algorithm.

3.1 Theoretical Analysis

SRKDA calculates the projective functions through the linear equations system in Eqn. (11). 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. (6) with respect to the eigenvalue 1. In this case, SRKDA is equivalent to ordinary KDA. Thus, it is interesting and important to see when the positive semi-definite kernel matrix K will be positive definite.

One of the most popular kernels is the Gaussian RBF kernel, $\mathcal{K}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$. Our discussion in this section will only focus on Gaussian kernel. Regarding the Gaussian kernel, we have the following lemma:

Lemma 2 (Full Rank of Gaussian RBF Gram Matrices [10]) *Suppose that $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m$ are distinct points, and $\sigma \neq 0$. The matrix K given by*

$$K_{ij} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2/2\sigma^2)$$

has full rank.

Proof See [10] and Theorem 2.18 in [14].

In other words, the kernel matrix K is positive definite (provided no two $\mathbf{x}_i$ are the same).

Thus, we have the following theorem:

Theorem 3 *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. (6) with respect to eigenvalue 1 when $\delta = 0$. In other words, the SRKDA and ordinary KDA are equivalent.*

Proof This theorem can be easily proofed by combining Lemma 2 and Theorem 1.

It is easy to check that the values of the i-th and j-th entries of any vector $\mathbf{y}$ in the space spanned by $\{\mathbf{y}_k\}$ in Eqn. (13) are the same as long as $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same class. Thus the i-th and

j-th rows of $\bar{Y}$ are the same, where $\bar{Y} = [\bar{\mathbf{y}}_1, \cdots, \bar{\mathbf{y}}_{c-1}]$. Theorem (1) shows that when the kernel matrix is positive definite, the $c-1$ projective functions of KDA are exactly the solutions of the $c-1$ linear equations systems $K\boldsymbol{\alpha}_k = \bar{\mathbf{y}}_k$. Let $\Theta = [\boldsymbol{\alpha}_1, \cdots, \boldsymbol{\alpha}_{c-1}]$ be the KDA transformation matrix which embeds the data points into the KDA subspace as:

$$\Theta^T[K(:, \mathbf{x}_1), \cdots, K(:, \mathbf{x}_m)] = \bar{Y}^T.$$

The columns of matrix $\bar{Y}^T$ are the embedding results of data samples in the KDA subspace. Thus, the data points with the same label are corresponding to the same point in the KDA subspace when the kernel matrix K is positive definite.

These projective functions are optimal in the sense of separating training samples with different labels. However, they usually overfit the training set thus may not be able to perform well for the test samples, thus the regularization is necessary.

3.2 Computational Analysis

The computation of SRKDA involves two steps: responses ($\bar{\mathbf{y}}_k$ in Eqn. 14) 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 [15].

To solve the $c-1$ linear equations systems in Eqn. (11), 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 [15]. With this Cholesky decomposition, the $c-1$ linear equations can be solved within m^2c flam [15]. Besides solving the SRKDA optimization problem, we also need to compute the kernel matrix K which requires $O(m^2n)$ flam, where n is the number of features. Thus, the computational cost of SRKDA is

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

which can be simplified as

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

Comparing to the computational cost of ordinary KDA in Eqn. (9), 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 provides us the chance to develop incremental version of SRKDA.

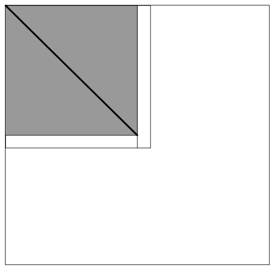
Figure 1: Sherman's march (Cholesky decomposition)

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 [15]. Actually, *Sherman's march*, one of the most popular Cholesky decomposition algorithms, is implemented in the incremental manner [15].

The procedure of Sherman's march is illustrated graphically in Figure 1. The gray area represents the part of the Cholesky decomposition that has already been computed with R and R^T separated by a diagonal line[1]. The white area represents untouched elements of the original matrix. The thin vertical box represents the column of R about to be computed. The algorithm is easy to derive. We show how to proceed from $(m-1)\times(m-1)$ submatrix 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 [15].

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 [15].

[1]Actually, we only need to store R.

Table 1: Computational complexity of KDA and SRKDA

Algorithm		operation counts (*flam* [15])
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

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

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

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

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

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 in both batch and incremental manners. All of our experiments have been performed on a P4 3.20GHz Windows XP machines with 2GB memory.

Table 2: Statistics of the three data sets

dataset	dim (n)	train size (m)	test size	# of classes (c)
Isolet	617	6238	1559	26
USPS	256	7291	2007	10
PIE	1024	8000	3554	68

5.1 Datasets

Three datasets are used in our experimental study, including spoken letter, handwritten digit image, and face image data sets. The important statistics of three datasets are summarized below (see also Table 2):

- The Isolet spoken letter recognition database[2] was first used in [6]. It contains 150 subjects who spoke the name of each letter of the alphabet twice. The speakers are grouped into sets of 30 speakers each, and are referred to as isolet1 through isolet5. In the past usage [6][5], isolet1&2&3&4 were used as the training set and isolet5 was used as the test set. For the purposes of our experiment, we also choose isolet5 as the test set and perform several runs with isolet1, isolet1&2, isolet1&2&3, and isolet1&2&3&4 as the training set respectively.
- The USPS handwritten digit database is described in [9]. A popular subset [3] contains 9298 16×16 handwritten digit images in total, which is then split into 7291 training images and 2007 test images. In our experiment, we train all the algorithms on the first 1500 (3000, 4500, 6000, and 7291) images in the training set and test on the 2007 test images.
- The CMU PIE face database[4] 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[5]. 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.

5.2 Compared algorithms

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

[2]http://www.ics.uci.edu/∼mlearn/MLSummary.html

[3]http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/datasets/multiclass.html#usps

[4]http://www.ri.cmu.edu/projects/project_418.html

[5]http://ews.uiuc.edu/∼dengcai2/Data/data.html

Table 3: Performance comparisons on Isolet dataset

	Error (%)					Time (s)					Speedup
Training Set	LDA	KDA	SRKDA	KDA/QR	SVM	LDA	KDA	SRKDA	KDA/QR	SVM	
Isolet1	15.27	**11.74**	**12.89**	17.19	**12.51**	1.93	18.86	1.21	0.93	4.75	15.6
Isolet1+2	6.61	**3.79**	**3.85**	7.63	4.11	2.14	134.6	5.51	3.60	13.79	24.4
Isolet1+2+3	5.90	**2.99**	**3.08**	7.12	3.34	2.37	451.6	14.09	7.98	23.84	32.1
Isolet1+2+3+4	5.71	**2.82**	**2.89**	6.86	3.27	2.56	991.2	27.86	14.02	34.82	35.6

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

Table 4: Performance comparisons on USPS dataset

	Error (%)					Time (s)					Speedup
Training Size	LDA	KDA	SRKDA	KDA/QR	SVM	LDA	KDA	SRKDA	KDA/QR	SVM	
1500	10.61	6.58	**5.88**	10.86	6.85	0.21	14.97	0.92	0.66	0.78	16.3
3000	9.77	**5.53**	**5.38**	10.66	**5.58**	0.27	111.9	4.35	2.61	2.20	25.7
4500	9.52	5.53	**4.88**	9.67	5.13	0.34	354.3	11.29	5.85	4.06	31.4
6000	9.92	5.03	**4.43**	9.37	5.08	0.40	825.3	22.74	10.41	6.22	36.3
7291	10.26	4.83	**4.04**	9.02	4.83	0.47	1553.6	37.59	15.60	8.18	41.3

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

Table 5: 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).

1. Linear Discriminant Analysis (LDA) [7], which provides us a baseline performance of linear algorithms. We can examine the usefulness of kernel approaches by comparing the performance of KDA and LDA.

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 [18], a KDA variation in which QR decomposition is applied rather than eigen-decomposition. Thus, KDA/QR is very efficient.

5. Support Vector Machine (SVM) [17], 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. SVM is used to get the sense that how good the performance of KDA is.

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.

5.3 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 ($3 \sim 5$) respectively.

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 in all the three data sets (even better than SVM). For Isolet data set, previous study [5] reported the minimum error rate training on Isolet1+2+3+4 by OPT[6] with 30 bit ECOC is 3.27%. KDA (SRKDA) achieved better performance in our experiment for this train/test split. For USPS data set, previous studies [14] reported error rate 3.7% for KDA and 4.0% for SVM, slightly better than the results in our experiment. For all the cases, KDA (SRKDA) achieved significantly better performance than LDA, which suggests the effectiveness of kernel approaches.

- 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. The time of training SRKDA is comparable with that of training SVM. SRKDA is faster than SVM on Isolet and PIE data sets, while slower than SVM on USPS data set. This is because the time of training SVM is dependant with the number of support vectors [2]. For some data sets with lots of noise (*e.g.*, USPS), the number of support vectors is far less than the number of samples. In this case, SVM can be trained very fast.

- The KDA/QR algorithm is very efficient because it only need to perform QR decomposition on matrices with size $m \times c$ [18]. However, there is no theoretical relation between the optimization problem solved in KDA/QR and that of the KDA. In all the three data sets, the performances of KDA/QR is the worst.

[6]Conjugate-gradient implementation of back-propagation

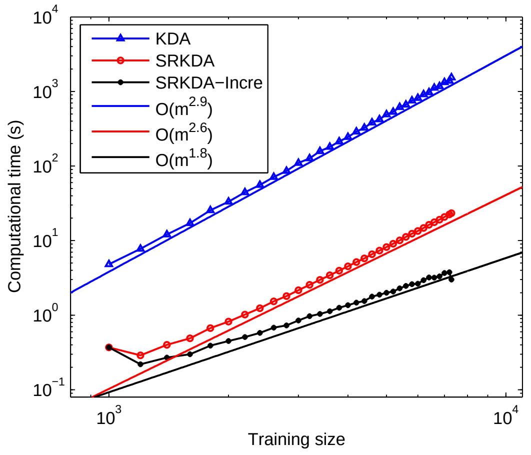

Figure 2: Computational cost of KDA, batch SRKDA and incremental SRKDA on the USPS data set.

5.4 Experiments on Incremental SRKDA

In this experiment, we study the computational cost of SRKDA performing in the incremental manner. The USPS and PIE data sets are used. We start from the training set with the size of 1000 (the first 1000 samples in whole training set) and increase the training size by 200 for each step. SRKDA is then performed in the incremental manner. It is important to note that SRKDA in the incremental manner give the exactly same projective functions as the SRKDA in the batch mode. Thus, we only care about the computational costs in this experiment.

Figure 2 and 3 shows log-log plots of how CPU-time of KDA (SRKDA, incremental SRKDA) increases with the size of the training set on USPS and PIE data set respectively. Lines in a log-log plot correspond to polynomial growth $O(m^d)$, where d corresponds to the slope of the line. The ordinary KDA scales roughly $O(m^{2.9})$, which is slightly better than the theoretical estimation. SRKDA in the batch mode has better scaling, which is also better than theoretical estimation with roughly $O(m^{2.6})$ over much of the range. This explains why SRKDA can be more than 27 times faster than ordinary KDA in the previous experiments. The SRKDA in the incremental mode has the best scaling, which is (to some surprise) better than quadratic with roughly $O(m^{1.8})$ over much of the range.

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

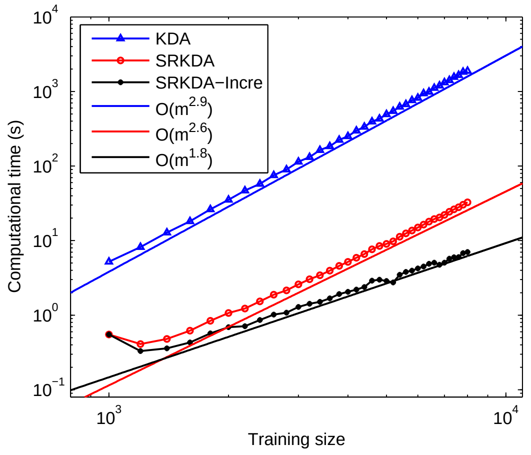

Figure 3: Computational cost of KDA, batch SRKDA and incremental SRKDA on the PIE data set.

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] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[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] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Jounal of Artificial Intelligence Research*, (2):263–286, 1995.

[6] M. A. Fanty and R. Cole. Spoken letter recognition. In *Advances in Neural Information Processing Systems 3*, 1990.

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

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

[9] J. J. Hull. A database for handwritten text recognition research. *IEEE Trans. Pattern Anal. Mach. Intell.*, 16(5), 1994.

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

[11] S. Mika, G. Rätsch, and K.-R. Müller. A mathematical programming approach to the kernel fisher algorithm. In *Advances in Neural Information Processing Systems 14*, 2000.

[12] 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.

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

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

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

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

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

[18] 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.