Margin Based Dimensionality Reduction and Generalization

Jing Peng*,[1], Stefan Robila[1], Wei Fan[2] and Guna Seetharaman[3]

[1]*Computer Science Department, Montclair State University, Montclair, NJ 07043, USA*

[2]*IBM T.J. Watson Research, Hawthorne, NY 10532, USA*

[3]*Computing Technology Applications Branch, Air Force Research Laboratory, Ohio, USA*

Abstract: Linear discriminant analysis (LDA) for dimension reduction has been applied to a wide variety of problems such as face recognition. However, it has a major computational difficulty when the number of dimensions is greater than the sample size. In this paper, we propose a margin based criterion for linear dimension reduction that addresses the above problem associated with LDA. We establish an error bound for our proposed technique by showing its relation to least squares regression. In addition, there are well established numerical procedures such as semi-definite programming for optimizing the proposed criterion. We demonstrate the efficacy of our proposal and compare it against other competing techniques using a number of examples.

Keywords: Dimensionality reduction, linear discriminent analysis, margin criterion, semi-definite programming, small sample size problem.

1. INTRODUCTION

In classification, many features or attributes often make the design of a classifier difficult and degrade its performance. This is particularly pronounced when the number of examples is small relative to the number of features. This fact is due to the curse of dimensionality. It states in simple terms that the number of examples required to properly compute a classifier grows exponentially with the number of features. For example, assuming features are correlated, approximating a binary distribution in a n dimensional feature space requires estimating $O(2^n)$ unknown variables [1]. In such situations, the problem often becomes intractable. This calls for reducing the number of features in constructing classifiers.

There are many dimensionality reduction techniques in the literature. The two most popular ones are principal components analysis (PCA) and linear discriminant analysis (LDA) [2]. Both techniques have been successfully applied to a wide variety of practical problems. By projecting data onto a linear subspace spanned by principal components, PCA achieves dimension reduction with the minimal data reconstruction error. On the other hand, without taking into account class information PCA cannot compute discriminant information required by classifiers. In this pape, we are concerned with LDA.

In LDA, we are given a set of l examples:

$$z = \{(x_i, y_i)\}_{i=1}^{l}. \tag{1}$$

These examples are independently and identically distributed (i.i.d.) from the probability space $Z = X \times Y$. Here probability measure ρ is defined but unknown, $x_i \in X \subset \Re^q$ are the q-dimensional inputs, and $y_i \in Y = [-M, M] \subset \Re$ are scalar labels. According to Fisher's criterion, one has to find a projection matrix $W \in \Re^{q \times d}$ that maximizes:

$$J(W) = \frac{|W^T S_b W|}{|W^T S_w W|} \tag{2}$$

where S_b and S_w are so-called between-class and within-class matrices, and d denotes the dimensions of the reduced space. In practice, the "small sample size" (SSS) problem is often encountered, when $l < q$. In this case S_w is singular. Therefore, the maximization problem can be difficult to solve.

To address this issue, the term εI is added, where ε is a small positive number and I the identity matrix of proper size. This results in maximizing

$$J(W) = |W^T S_b W| / |W^T (S_w + \varepsilon I) W|. \tag{3}$$

It can then be solved without any numerical problems. This is a special case of Friedman's regularized discriminant analysis with regard to the small sample size problem [3]. In [4], it is shown that naive Bayes outperforms LDA under broad conditions. In this work, we address this problem in the context of dimensionality reduction.

In this paper, we present a margin based criterion for dimensionality reduction that potentially provides a solution to the problem implied by the above discussion. In particular, we show that

- our margin based criterion for dimensionality deduction is closely related to the average margin criterion [5];
- our objective does not involve the inverse of S_w and can be optimized using algorithms such as semi-definite programming, thereby avoiding the small sample size problem; and

*Address correspondence to this author at the Computer Science Department, Montclair State University, Montclair, NJ 07043; USA; Tel: 973-655-7975; Fax: 973-655-4164; E-mail: peng@pegasus.montclair.edu

- we establish an error bound for the proposed technique.

We demonstrate the efficacy of our proposed technique using a variety of examples. We note that this work extends significantly in terms of theoretical analysis and experimental evaluation of our earlier work that appeared in the proceedings of the IEEE International Conference on Data Mining 2007 [6].

The rest of the paper is organized as follows. Section 2 provides a discussion on related work in discriminant analysis and subspace techniques. Section 3 introduces our proposal on discriminant analysis that derives linear discriminants in two class problems by optimizing a weighted additive criterion. Section 4 shows a procedure that optimizes our criterion and its correctness. Section 5 establishes an error bound for the proposed technique by showing its relationship to regularized least squares. Section 6 demonstrates how to extend our criterion to the multi-class case. Section 7 presents experimental evaluation of the proposed technique against several competing techniques using a variety of real data sets. Finally, Section 8 summaries our work and points out future research directions.

2. RELATED WORK

A number of proposals has been introduced to address the computational difficulty associated with LDA when the small sample size problem occurs (S_w becomes singular). A straightforward method is to use the pseudo-inverse of S_w^+ in place of S_w^{-1}. While simple, the method does not guarantee that Fisher's objective will be optimized by the eigenvector matrix of $S_w^+ S_b$. Furthermore, computing S_w^+ itself is ill posed. Another simple method is to first use PCA to remove the null space of S_w, and then apply LDA to the reduced representation. Fisherface is one such example [7]. However, this method remains sub-optimal because the null space of S_w potentially contains discriminant information [8].

Another technique, newLDA [8], first transforms the data into the null space of S_w. It then applies PCA to maximize the between-class scatter matrix in the transformed space. While newLDA mitigates the small sample size problem to the extent possible, its performance degrades with decreasing dimensions of the null space. A variant of LDA+PCA is proposed in [9]. The method first discards the null space of $S_w + S_b$ that is the common null space of both S_w and S_b. And as such, discarding this null space does not lose any discriminant information. The method then applies LDA+PCA to the reduced representation in the transformed space. A direct LDA (DLDA) is a method that throws away the the null space of S_b [10]. If $S_w + S_b$ replaces S_w, DLDA reduces to PCA+LDA [10].

Discriminant analysis based on the average margin is proposed in [5]. The technique does not involve inverting matrices, thereby avoiding the small sample size problem. This technique is closely related to our proposal, as we shall see later.

Recently, a dimension reduction technique, called linear feature extraction (LFE) is introduced in [11]. Let x be an instance. We define the *near hit* or nh of x as its nearest neighbor that comes from the same class as x. Similarly, we define the *near miss* or nm as the nearest neighbor of x that comes from the opposite class. Then the hypothesis margin of x with respect to labeled data L is defined as [12]

$$\sigma(x) = \| x - nm(x) \| - \| x - nh(x) \| . \tag{4}$$

The hypothesis margin is easy to compute and lower bounds the sample margin [12].

Let $h(x) = x - nh(x)$ and $m(x) = x - nm(x)$. We define two matrices, near hit S_h and near miss S_m, as follows.

$$S_h = \sum_{i=1}^{l} h(x_i) h(x_i)^t \tag{5}$$

and

$$S_m = \sum_{i=1}^{l} m(x) m(x)^t . \tag{6}$$

Instead of optimizing the margin (4) by selecting features, a technique described in [11] computes a linear transform that optimizes the following

$$\max_x \quad x^t (S_m - S_h) x \tag{7}$$
$$(x^t x)^2 = 1$$

which is very similar to our criterion (15). To be less sensitive to noise, k near misses and hits are often used in practice to optimize the margin for some integer k [11].

We can rewrite S_h as follows

$$\begin{aligned} S_h &= \sum_{x \in \{+1\}} (x - m_{+1})(x - m_{+1})^t + \sum_{x \in \{-1\}} (x - m_{-1})(x - m_{-1})^t \qquad (8) \\ &= \sum_{x \in \{+1,-1\}} x x^t - 2 \sum_{x \in \{+1\}} x m_{+1}^t + \sum_{x \in \{+1\}} m_{+1} m_{+1}^t + \\ &\quad -2 \sum_{x \in \{-1\}} x m_{-1}^t + \sum_{x \in \{-1\}} m_{-1} m_{-1}^t \\ &= \sum_{x \in \{+1,-1\}} x x^t - \frac{l}{2} m_{+1} m_{+1}^t - \frac{l}{2} m_{-1} m_{-1}^t \end{aligned}$$

Similarly, S_m can be written as

$$\begin{aligned} S_m &= \sum_{x \in \{+1\}} (x - m_{-1})(x - m_{-1})^t + \sum_{x \in \{-1\}} (x - m_{+1})(x - m_{+1})^t \qquad (9) \\ &= \sum_{x \in \{+1,-1\}} x x^t - l m_{+1} m_{-1}^t - l m_{+1} m_{-1}^t + \frac{l}{2} m_{+1} m_{+1}^t + \frac{l}{2} m_{-1} m_{-1}^t \end{aligned}$$

Then

$$\begin{aligned} S_m - S_h &= l (m_{+1} m_{+1}^t - 2 m_{+1} m_{-1}^t + m_{-1} m_{-1}^t) \qquad (10) \\ &= l S_b . \end{aligned}$$

This shows that when near hits and near misses are extended to the entire neighborhood, maximizing the margin reduces to maximizing the between-class scatter matrix. Because it ignores the within-class scatter matrix, it cannot be optimal. This lends theoretical support to the practical observation that the average neighborhood for near hit and near miss in Relief should be somewhere between 1 and $l/2$ [11].

A metric space dimension reduction technique, called discriminant neighborhood embedding (NDE), is introduced in [13]. The idea is to find a linear transform such that in the transformed space total within class distance is minimized, while total between class distance is maximized. Let $x_j \in NB_w(x_i)$ if x_j is a within class neighbor of x_i, and $x_j \in NB_b(x_i)$ if x_j is a between class neighbor of x_i. The neighborhood can be computed using k nearest neighbors. Then this objective is accomplished by defining an adjacency matrix F, where

$$F_{ij} = \begin{cases} 1 & if\ x_i \in NB_w(x_j)\ or\ x_j \in NB_w(x_i); \\ -1 & if\ x_i \in NB_b(x_j)\ or\ x_j \in NB_b(x_i); \\ 0 & otherwise. \end{cases}$$

The objective is to find P such that

$$tr(P^t X(S-F)X^t P)$$

is minimized, subject to $P^t P = I$. Here X is the data matrix, and S is a diagonal matrix, where $S_{ii} = \sum_j F_{ij}$ or $S_{ii} = \sum_j F_{ji}$.

If we use $k = 1$ to compute NB_w and NB_b, we can write $X(S-F)$ as

$$M = (mh(x_1)\cdots mh(x_l)),$$

where $mh(x_i) = nm(x_i) - nh(x_i)$ represents the difference between the near miss $nm(x_i)$ and the near hit $nh(x_i)$ of x_i, respectively [12]. In this case, the objective becomes maximizing $tr(P^t MX^t P)$ over P, subject to $P^t P = I$. This is in many ways similar to the idea presented in [14] where we compute a subspace by pooling local information $(nm(x_i) - nh(x_i))x_i^t$, for $i = 1,\cdots,l$. Here the local information is the cross covariance of an instance and the difference between its near miss and near hit. If we compute NB_w and NB_b over the entire classes, i.e., $k = l/2$ (assuming each class has the same number of examples), it can be shown that

$$X(S-F)X^t = S_b.$$

The result is similar to LFE Eq. (11). In practice, k is chosen somewhere between 1 and $l/2$.

Several techniques have recently been proposed to improve the Fisher criterion in heteroscedastic data [15, 16]. These techniques employ Chernoff distance to capture difference in variance between within class matrices. Like the Fisher criterion, it involves computing the inverse of within class matrices. Thus, it potentially suffers from the small sample size problem. In this work, we are mainly interested in addressing the small sample size problem facing LDA.

3. A MARGIN CRITERION FOR DISCRIMINANT ANALYSIS

In this section, we first review LDA using Fisher's criterion, and then go on to investigate discriminant analysis using a margin based criterion and related optimization techniques.

3.1. Linear Discriminant Analysis

In LDA, within-class, between-class, and mixture scatter matrices are used to formulate the criteria of class separability. Consider a J class problem, where m_0 is the mean vector of all data, and m_j is the mean vector of j th class data. A within-class scatter matrix characterizes the scatter of samples around their respective class mean vectors, and it is expressed by

$$S_w = \sum_{j=1}^{J} p_j \sum_{i=1}^{l_j} (x_i^j - m_j)(x_i^j - m_j)^T, \tag{11}$$

where l_j is the size of the data in the j th class and p_j ($\sum_j p_j = 1$) represents the proportion of the j th class contribution. A between-class scatter matrix characterizes the scatter of the class means around the mixture mean m_0. It is expressed by

$$S_b = \sum_{j=1}^{J} p_j (m_j - m_0)(m_j - m_0)^T. \tag{12}$$

The mixture scatter matrix is the covariance matrix of all samples, regardless of their class assignment, and it is given by

$$S_m = \sum_{i=1}^{l} (x_i - m_0)(x_i - m_0)^T = S_w + S_b. \tag{13}$$

The Fisher criterion is used to find the projection matrix that maximizes the objective (2). In order to determine the matrix W that maximizes $J(W)$, one can solve the generalized eigenvalue problem: $S_b w_i = \lambda_i S_w w_i$. The eigenvectors corresponding to the largest eigenvalues form the columns of W. For a two class problem, it can be written in a simpler form: $S_w w = m = m_1 - m_2$, where m_1 and m_2 are the means of the two classes.

3.2. Margin Criterion for Linear Dimensionality Reduction

Here we first focus on two class problems. The multi-class case will be discussed later. The goal of LDA is to find a direction w that simultaneously places two classes afar and minimizes within class variations. Fisher's criterion (2) achieves this goal. Alternatively, we can achieve this goal by maximizing

$$J(w) = tr(w^t(\lambda S_b - S_w)w), \tag{14}$$

where tr denotes the trace operator, and $\lambda > 0$ is a constant that weighs relative importance of the two terms S_b and S_w in determining the outcome of linear discriminants. Large λ values ignore within class spread, while small λ values penalize discriminants that result in large within class variations.

Notice that $tr(S_b)$ measures the overall scatter of class means. Therefore, a large $tr(S_b)$ implies that the class means spread out in a transformed space. On the other hand, a small $tr(S_w)$ indicates that in the transformed space the spread of each class is small. Thus, when maximized, J indicates that data points are close to each other within a class, while they are far from each other if they come from different classes.

To see our proposal Eq. (14) is margin based, notice that maximizing $tr(S_b - S_w)$ is equivalent to maximizing $J = \frac{1}{2}\sum_i^2 \sum_j^2 p_i p_j d(C_i, C_j)$, where p_i denotes the probability of class C_i. The interclass distance d is defined as $d(C_i, C_j) = d(m_i, m_j) - tr(S_i) - tr(S_j)$, where m_i represents the mean of class C_i, and S_i represents the scatter matrix of class C_i. As noted in [5], $d(C_i, C_j)$ measures the average margin between two classes. Therefore, maximizing our objective produces large margin linear discriminants. In addition, there is no need to calculate the inverse of S_w, thereby avoiding the small sample size problem associated with the Fisher criterion.

4. COMPUTING LINEAR DISCRIMINANTS WITH SEMI-DEFINITE PROGRAMMING

Suppose that w optimizes (14). So does cw for any constant $c \neq 0$. Thus we require that w have unit length. The optimization problem then becomes

$$\max_{w} \quad tr(w^t(\lambda S_b - S_w)w)$$

$$\textit{subject to:} \quad \| w \| = 1.$$

This is a constraint optimization problem. Since $tr(w^t(\lambda S_b - S_w)w) = tr((\lambda S_b - S_w)ww^t) = tr((\lambda S_b - S_w)X)$, where $X = ww^t$, we can rewrite the above constraint optimization problem as

$$\begin{aligned} \max_{X} \quad & tr((\lambda S_b - S_w)X) \\ & I \bullet X = 1 \\ & X \geq 0 \end{aligned} \qquad (15)$$

where I is the identity matrix and the inner product of symmetric matrices is $A \bullet B = \sum_{i,j}^n a_{ij} b_{ij}$, and $X \geq 0$ means that the symmetric matrix X is positive semi-definite. Indeed, if X is a solution to the above optimization problem, then $X \geq 0$ and $I \bullet X = 1$ implies $\| w \| = 1$, assuming rank(X) = 1.

The above problem is a semi-definite program (SDP), where the objective is linear with linear matrix inequality and affine equality constraints. Because linear matrix inequality constraints are convex, SDPs are convex optimization problems. The significance of SDP is due to several factors. SDP is an elegant generalization of linear programming, and inherits its duality theory. For a comprehensive overview on SDP, see [17].

SDPs arise in many applications, including sparse PCA, learning kernel matrices, Euclidean embedding, and others. In general, generic methods are rarely used for solving SPDs, because their time grows at the rate of $O(n^3)$ and their memory grows in $O(n^2)$, where n is the number of rows (or columns) of a semidefinite matrix. When n is greater than a few thousands, SDPs are typically not used. However, there are algorithms that have a good theoretical foundation to solve SDPs [17]. In addition, semidefinite programming is a very useful technique for solving many problems. For example, SDP relaxations can be applied to clustering problems such that after solving a SDP, final clusters can be computed by projecting the data onto the space spanned by the first few eigenvectors of the SDP solution. For large-scale problems, there is a tremendous opportunity for exploiting special structures in problems, as those suggested in [18, 19].

Assume rank (X) = 1. Since X is symmetric, one can show that rank (X) = 1 iff $X = ww^t$ for some vector w. Therefore, we can recover w from X as follows. Select any column (say the i th column) of X such that $X(1,i) \neq 0$, and let

$$w = X(:,i)/X(1,i), \qquad (16)$$

where $X(:,i)$ denotes the i th column of the matrix X. Thus, our goal here is to ensure the solution X to the above constraint optimization problem has rank at most 1.

One way to guarantee rank (X) = 1 is to use rank (X) = 1 as an additional constraint in the optimization problem. However, the constraint rank(X) = 1 is not convex and the resulting problem is difficult to solve. It turns out that the above formulation (15) is sufficient to ensure that the rank of the optimal solution X to Eq. (15) is one, i.e., rank (X) = 1.

Theorem 1 Let X be the solution to the semi-definite program (15). Also, let $rank(X) = r$. Then $r = rank(X) = 1$.

Proof. We rewrite $\lambda S_b - S_w = 2\lambda S_b - S_m$, where $S_m = S_b + S_w$. Let $null(A)$ denote the null space of matrix A. Since $null(S_m) \subseteq null(S_b)$, there exists a matrix $P \in \Re^{q \times s}$ that simultaneously diagonalizes S_b and S_m [20], where $s \leq \min\{l-1, q\}$ is the rank of S_m.

The matrix P is given by

$$P = Q\Lambda_m^{-1/2}U,$$

where Λ_m and Q are the eigenvalue and eigenvector matrices of S_m, and U is the eigenvector matrix of $\Lambda_m^{-1/2}Q^t S_b Q \Lambda_m^{-1/2}$. Thus, the columns of P are the eigenvectors of $2\lambda S_b - S_m$ and the corresponding eigenvalues are $2\lambda\Lambda_b - I$. We then have

$$P^tS_bP = \Lambda_b, \quad P^tS_mP = I. \tag{17}$$

where $\Lambda_b = diag\{\sigma_1, \cdots, \sigma_s\}$.

Consider the range of P over $Y \in \Re^{s \times q}$ with $rank(Y) = s$. The range $W = PY$ includes all $q \times q$ matrices with $rank = s$. Then

$$\max_W tr(W^t(2\lambda S_b - S_m)W) = \max_Y tr((PY)^t$$
$$(2\lambda S_b - S_m)PY) = \max_Y tr(Y^t(2\lambda\Lambda_b - I)Y).$$

It is straightforward to show that the maximum is attained by $Y = [e_1e_2 \cdots e_r; 0]$, where e_i is a vector whose i th component is one and the rest is 0. From this it is clear that $W = PY$ consists of the first r columns of P, i.e., the eigenvectors corresponding to $2\lambda\sigma_i - 1 > 0$.

Now, since $X = WW^t$, we have $X = \sum_{i=1}^{r} w_iw_i^t$. Thus,

$$tr(X) = \sum_{i=1}^{r} w_i^tw_i = r.$$

However, the constraint $I \cdot X = 1$ states that $tr(X) = 1$. It follows that $r = 1$. That is, $rank(X) = 1$. Therefore, our procedure for computing w from the matrix X Eq. (16) is guaranteed to produce the correct answer. We call our algorithm SDP-LDA.

While the criterion Eq. (14) is different from the Fisher criterion Eq. (2), it is very competitive[1]. Here we use the Iris data to show that Eq. (14) is very competitive. In this example, all three classes of the Iris data are used, where each class has 50 examples. We randomly choose 60% as training and the remaining 40% as testing. Since S_w is non-singular, the small sample size does not occur. Since there are three classes, we use two linear discriminants computed according to the Fisher criterion to project the data. For the proposed criterion $\lambda S - b - S_w$ Eq. (14) we use one discriminant to represent the data in the reduced space. One nearest neighbor rule is used to predict the class label in the reduced space. The average accuracies over 200 runs are 0.9634 Eq. (14) and 0.9510 (Fisher), respectively. This example shows that Eq. (14) is indeed competitive against the Fisher criterion, even with less resources. We will see very similar results later in the experimental section.

5. ERROR BOUND

We establish an error bound for our learning algorithm in two steps. First, we show the relationship between our learning algorithm and regularized LDA Eq. (3). Second, we show that maximizing Eq. (3) produces the same solution as minimizing the regularized least squares

$$f = \arg\min_{f \in H} \frac{1}{l}\sum_{i=1}^{l}(y_i - f(x_i))^2 + \gamma \parallel f \parallel_H^2, \tag{18}$$

when the hypothesis space H is linear $H_L = \{f \mid f(x) = w^tx\}$. Here γ represents the regularization constant. In this case, we have $\parallel f \parallel_H^2 = w^tw$, and thus

$$w_{opt} = \arg\min_w \frac{1}{l}\sum_{i=1}^{l}(y_i - w^tx_i)^2 + \gamma w^tw. \tag{19}$$

We then use the bound for w_{opt} to bound our learning algorithm. The following lemma establishes a relationship between regularized LDA (3) and our proposal (15).

Lemma 2 *Let regularized LDA be defined by* (3). *Then solving* Eq. (15) *is a special case of solving regularized LDA in two class problems.*

Proof. Rewrite (15) as

$$J(w) = \frac{1}{2}tr(w^t(\lambda S_b - S_w)w) + \alpha(1 - tr(ww^t)), \tag{20}$$

where α is the Lagrangian multiplier. We then have

$$\frac{\partial J}{\partial w} = (\lambda S_b - S_w)w - \alpha w.$$

Setting the above to zero, we obtain

$$(S_w + \alpha I)w = \lambda S_bw. \tag{21}$$

Solving the above is equivalent to solving

$$(S_w + \alpha I)w = m_1 - m_2, \tag{22}$$

since S_bw is always in the direction $m_1 - m_2$ (scale factor for w has no consequence), where m_i ($i = 1,2$) represents the mean of class i. Thus, setting ε in Eq. (3) to α results in linear solutions to regularized LDA (3) in two class problems.

The above lemma states that our proposal (15) and regularized LDA (3) produce the same linear discriminants in two class problems. We now show in the following lemma that the solution to (3) is equivalent to the solution to (19).

Lemma 3 *The linear solution to the regularized Fisher's criterion* (3) *is equivalent to the linear solution* w_{opt} *to the least squares criterion* (19)*, up to a constant, in two class problems.*

The proof of the Lemma is given in Appendix A. Combining the above two lemmas, we conclude that

Theorem 4 *The linear solution to* Eq. (15) *is equivalent to the linear solution* w_{opt} *to the least squares criterion* (19)*, up to a constant, in two class problems.*

Recall that LDA tries to find a transformation f such that the transformed data has a large between class difference and small within class variation. The equivalence between Eq. (19) and Eq. (3) can be stated in another way: one minimizes the mean squared error with respect to each class mean while keeping the mean difference fixed.

We define f_ρ as the best function that minimizes the mean squared error. That is

[1] In [24], it is incorrectly shown that Eq. (14) is equivalent to the Fisher criterion Eq. (2).

$$f_\rho = \arg\min_f \int_Z (y - f(x))^2 d\rho. \tag{23}$$

Given the training data z Eq. (1), the goal of learning is to find a function f_z that comes as close as possible to f_ρ

$$f_{opt} = \arg\min_{f_z} \int_Z (f_z - f_\rho)^2 d\rho.$$

Here ρ is unknown and so is f_ρ. Instead, we consider empirical error minimization. However, solving empirical error minimization often leads to over fitting and the solution is unstable if no constraint is placed on f_z. Thus, we minimize the regularized empirical error Eq. (19).

The equivalence between our proposal and (19) allows us to establish an error bound for (15) in the following theorem, due to [21, 22]. First, let

$$L_k f(x) = \int_X f(x^{'}) k(x, x^{'}) d\rho_X, \tag{24}$$

where ρ_X is the marginal probability measure on X and $k(x, x^{'})$ is a continuous and symmetric, positive semi-definite function on $X \times X$. Throughout the paper, we assume that there exists a positive constant M that satisfies

$$| f(x) - y | \leq M \tag{25}$$

almost everywhere.

Theorem 5 *Let* z Eq. (1) *be randomly drawn according to* ρ*, and* f_ρ *be defined by* Eq. (23). *Then for any* $0 < \delta < 1$*, with confidence* $1 - \delta$ *the error bound for the solution* w_{opt} *of* Eq. (19), *thus* Eq. (15)*, is given by:*

$$\int (w_{opt} - f_\rho)^2 d\rho_X \leq S(\gamma) + A(\gamma), \tag{26}$$

where $A(\gamma)$ (approximation error in this context) and $S(\gamma)$ (sample error) are given by

$$A(\gamma) = \gamma^{1/2} \| L_k^{-\frac{1}{4}} f_\rho \|^2$$

and

$$S(\gamma) = \frac{32 M^2 (\gamma + C_k)^2}{\gamma^2} v^*(m, \delta),$$

where $v^*(m, \delta)$ is the unique positive solution of

$$\frac{m}{4} v^3 - \ln(\frac{4m}{\delta}) v - c = 0. \tag{27}$$

Here $C_k, c > 0$ depend only on X and k.

Decomposition $S(\gamma) + A(\gamma)$ represents the bias-variance tradeoff. Given a hypothesis space, $A(\gamma)$ measures the "error" between the optimal function learnable from the hypothesis space and the true target f_ρ. $S(\gamma)$ on the other hand bounds the sample error and is essentially derived by the law of large numbers.

We can bound $S(\gamma) + A(\gamma)$ by [23] (Corollary 5)

$$O((\frac{1}{m})^{1/4})$$

by taking

$$\gamma = (\frac{\log(4/\delta)^2}{m})^{\frac{1}{4}}. \tag{28}$$

This is accomplished primarily through the employment of a concentration inequality, while still following the essential outline of the approach described in [21]. The error bound is not only sharper, but also provides a guide to the asymptotic value of the regularization parameter γ Eq. (28). It shows that for a fixed δ, γ goes to zero as the number of training examples goes to infinity, as expected. For a given training data set, high confidence (low δ) requires large γ. Notice that there is no conflict between γ defined in [21] and the one in [23]. γ in [21] is optimal within the settings discussed in the paper.

6. MULTI-CLASS DLA

We have presented a margin based criterion as an alternative to Fisher's criterion. We have shown how to optimize our criterion with semi-definite programming to obtain the optimal linear transform in two class problems, where one dimensional projection is adequate. However, LDA is generally used to find a subspace with d dimensions for multiple class problems. In this section we extend our SDP approach to LDA to the multi-class case.

We start with $A_1 = \lambda S_b - S_w$, where S_b and S_w are computed as in the two class case. We solve the problem in (15)

$$\max_X \quad tr(A_1 X)$$
$$X \geq 0$$
$$I \bullet X = 1$$

to obtain the solution $X_1 = w_1 w_1^t$. Once we have obtained the solution $X_j = w_j w_j^t$, we inflate A_j to obtain

$$A_{j+1} = A_j + X_j,$$

from which we compute $X_{j+1} = w_{j+1} w_{j+1}^t$ for $j = 1, \cdots, C-1$, where C represents the number of classes. Here X_j s force w_{j+1} to be orthogonal to w_j s, as desired. To see this, we write

$$\begin{aligned} w_{j+1}^t (A_j) w_{j+1} &= w_{j+1}^t (A_1 + w_1 w_1^t + \cdots + w_j w_j^t) w_{j+1} \\ &= w_{j+1}^t A_1 w_{j+1} + (w_{j+1}^t w_1)^2 + \cdots + (w_{j+1}^t w_j)^2. \end{aligned}$$

Thus, $w_{j+1}^t (A_j) w_{j+1}$ is minimized when w_{j+1} minimizes A_1 and is orthogonal to the w_j s, since $w_j \neq 0$ for all j. Its complexity is at most C times the complexity for two class problems.

It can also be shown that the solution obtained as such is the same as the solution obtained by treating the multi-class problem as C binary problems, where the i th two class problem treats the i th class as one class and all remaining classes as the second class. Each binary class problem is solved first, and after finding all subspaces, PCA is applied to find eigenvectors having the largest eigenvalues, which are the solution of the original multi-class LDA problem.

7. EXPERIMENTS

In this section we compare the proposed technique with several competing subspace techniques using a number of examples, including multi-class facial and binary data sets.

7.1. Competing Methods

The following subspace techniques will be evaluated. All procedural parameters are determined through 10-fold cross validation.

SDP-LDA: Our proposed algorithm (15). To solve the semi-definite program, we used the general purpose optimization software SeDuMi [24].

PCA+LDA: Apply PCA to remove the null space of S_w first, then maximize [7, 25]

$$J(W) = | W^T S_b W | / | W^T S_w W |.$$

S-LDA: Same as PCA+LDA but maximizing [2, 26]: $J(W) = | W^T S_b W | / | W^T S_m W |$.

newLDA: If S_w is full rank then solve regular LDA; else in the null space of S_w, find the eigenvectors of S_b with largest eigenvalues [8].

DLDA: Apply PCA to remove the null space of S_b first, then find the eigenvectors of S_w corresponding to the smallest eigenvalues [10].

DNE: The discriminant neighborhood embedding algorithm presented in [13]. It finds a linear transform that maps within class samples closer together and between class data samples away from each other.

LFE: The linear feature extraction algorithm proposed in [11]. Similar to Relief, it finds a linear subspace by maximizing the hypothesis margin. Notice that LFE is a two class subspace technique. And as such, it is applied to the two class problems (Section 7.3).

C-LDA: The linear dimensionality reduction algorithm using the Chernoff criterion [15]. This method is applied to the two class data experiments only (Section 7.3), since it has some difficulty in computing the inverse of within class matrices on the two image data sets.

It should be noted that PCA+LDA and S-LDA can be equivalent when S_w and S_m span the same subspace. However, they are different when S_b totally or partially spans the null space of S_w, thus S_w and S_m span different subspaces. For face recognition the latter case turns out to be more common. In [8], Chen *et al.* show that the null space of S_w contains discriminant information. They also show that Scatter-LDA is not "optimal" in that it fails to distinguish the most discriminant information in the null space of S_w. Thus they propose the newLDA method. However, newLDA fell short of making use of any information outside of that null space.

In all the experiments, the data in a reduced space are normalized to have zero mean and unit variance along each dimension (i.e., Gaussian normalization). First, the mean and variance along each dimension are calulated using the training data. Then, the training mean and variance are used to normalize the test data.

7.2. Facial Images

7.2.1. Feret Face Data

The FERET face data [27] is now a standard facial database for testing and evaluating facial classification algorithms. Each image has 384×256 pixels. Sample images used in the experiments are shown in Fig. (**1**). The images used here involve variations in facial expressions and illumination.

For the FERET data, we extracted 150 images, where there are 50 individuals with three images from each. We randomly choose two images per person for training, and the remaining one for testing. Thus, for the FERET data we have 100 training and 50 test images. Each image is preprocessed to align along the eyes and reduced in size to 150×130 pixels. The corresponding preprocessed images are shown in Fig. (**2**). The preprocessed images of 150×130 pixels are first transformed into a space of 149 dimensions spanned by the 150 images through PCA. As a result, we are facing the challenge of the small sample size problem.

Fig. (1). Feret sample images.

Fig. (2). Normalized Feret sample images.

Subspaces are calculated from the training data, and the one nearest neighbor (NN) classifier is used to obtain accuracy after projecting the data onto the subspace. We prefer a simple classifier in order to highlight the subspace methods. To obtain average performance, each methods repeated 10 times. The average accuracy as a function of dimensionality is shown in Fig. (**3**).

The X-axis represents the dimensionality of the subspace. For each technique, the higher the dimension, the less discriminant the dimension. For most techniques, the accuracy rate increases quickly around the first 10 dimensions, and then increase slowly with additional dimensions.

SDP-LDA is uniformly better than any other algorithms on the Feret data, especially at lower dimensions, demonstrating its efficacy. It achieves the highest accuracy rate of 0.922 on the Feret data. newLDA performs quite well in the experiment, demonstrating that the most discriminant information is in the null space of S_w, for the facial recognition tasks. On the other hand, S-LDA does not perform well at lower dimensional subspaces. But it

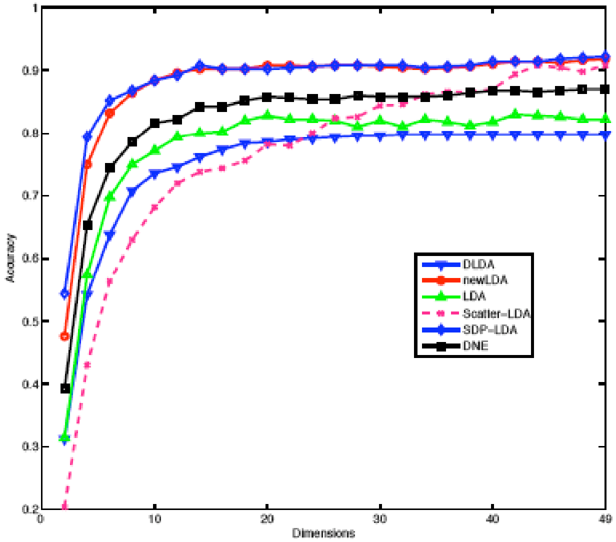

Fig. (3). Comparison of SDP-LDA, DLDA, LDA, newLDA, S-LDA, and DNE on the Feret image data.

eventually performs better than PCA+LDA, when the number of dimensions is large enough. All methods achieve higher accuracy rates toward higher dimensional subspaces, which is not surprising, for it is a 50 class problem. It should be noted that the performance of newLDA and S-LDA (its tail is not shown in the plot) drops quickly with unnecessary dimensions.

7.2.2. ORL Face Data

The ORL data set [28] is used in this experiment. The size of each image is 92×112 . We extracted 120 images, where there are 40 subjects with three images from each. Sample images are shown in Fig. (**4**). Similar to the Feret data, the images of 92×112 pixels are first transformed into a space of 119 dimensions spanned by the 120 images through PCA.

Fig. (4). ORL sample images.

We randomly choose two images per person for training, and the remaining one for testing. We have 80 training and 40 test images. Again, the one nearest neighbor classifier is used to obtain accuracy rates after projecting the data onto the subspace. To obtain average performance, each method is repeated 10 times. The average accuracy as a function of dimensionality is shown 5.

SDP-LDA is again uniformly better than any other algorithms on the ORL data. It achieves the highest accuracy rate of 0.8875 on ORL. For most techniques, the accuracy rate increases quickly around the first 10 dimensions, and then increase slowly with additional dimensions. The results are similar to what we observe on the Feret data.

7.3. Binary Data Sets

In these experiments, we compare the seven competing methods on a number of two class classification problems. We use 12 data sets from the UCI database and the cat and dog data (CatDog). They are all two class classification problems. The cat and dog data set is composed of two hundred images of cat and dog faces. There are equal number of cats and dogs in the data set. Each image is a black-and-white 64×64 pixel image, and the images have been registered by aligning the eyes. PCA is first applied to reduce the number of dimensions from 4096 to 199.

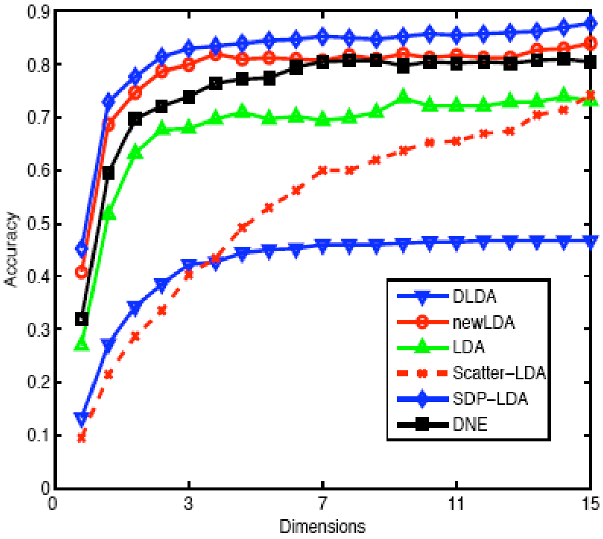

Fig. (5). Comparison of SDP-LDA, DLDA, LDA, newLDA, S-LDA, and DNE on the ORL image data.

For each data set, we randomly choose 60% as training and the remaining 40% as testing. We train the seven methods on the training data and obtain a one-dimensional subspace. We then project both training and test data on the chosen subspace and use the 1NN classifier to obtain error rates. Note that for the two class case, one dimensional subspace is sufficient. Again, all procedural parameters for all the methods are chosen through 10-fold cross-validation. We repeat the experiments 30 times on each data set to obtain the average accuracy rates.

The average accuracy rates over 30 runs are shown in Table **1**. Both SDP-LDA and S-LDA (Scatter-LDA) come out first five times out of the 13 data sets, while each of the Chernoff LDA, LFE and DNE methods comes out first once. The differences are significant only on three data sets: New Thyroid, Sonar, and Cat and Dog (pair-t test with a 95% confidence level). Overall, three methods: SDP-LDA, S-LDA and C-LDA are very similar on the binary data sets. Another way to look at these methods is to see how they perform across tasks. That is, we want to see how well or robust a method can perform when a given task is not in favor of this particular method. The following can be used to measure robustness. For each method m we compute the ratio b_m between its error rate e_m and the smallest error rate over all methods being compared in a particular example:

$$b_m = e_m / \min_{1 \le k \le 13} e_k .$$

One can see that b_m for the most robust method are centered around one with zero spread.

Fig. (**6**) plots the distribution of b_m for each method over the 13 data sets. The box area represents the lower and upper quartiles of the distribution that are separated by the median.

Table 1. Classification Accuracy Rates in Subspaces Computed by the 8 Competing Methods Using 1NN Classifier, on the 13 Data Sets

Data Set	DLDA	PCA+LDA	newLDA	SDP-LDA	S-LDA	LFE	DNE	C-LDA
BreastCancer	0.6541	0.6445	0.6445	0.6536	0.6659	0.6386	0.6327	0.6609
Credit	0.7969	0.7488	0.5952	0.8062	0.8162	0.8088	0.8004	0.8013
Heart Cleve	0.7551	0.7530	0.7530	0.7644	0.7780	0.7581	0.7466	0.7695
Heart Hun	0.7607	0.7440	0.7077	0.7726	0.7679	0.7526	0.7641	0.7628
Ionosphere	0.7604	0.7514	0.7514	0.8421	0.8282	0.7379	0.7268	0.8364
Letters	0.9482	0.9379	0.9379	0.9692	0.9683	0.9530	0.9521	0.9686
Thyroid	0.8203	0.7936	0.7936	0.8221	0.8233	0.9634	0.9529	0.8267
Pima	0.6564	0.6831	0.6831	0.6824	0.6897	0.6728	0.6661	0.6883
Glass	0.9165	0.8588	0.8588	0.9059	0.9182	0.9082	0.9035	0.9153
Iris	0.8700	0.9362	0.9362	0.9437	0.9437	0.8912	0.8988	0.9450
Cancer Wis	0.9596	0.9559	0.9559	0.9577	0.9551	0.9557	0.9605	0.9550
Sonar	0.6774	0.5146	0.5030	0.7122	0.6896	0.6744	0.6659	0.6848
CatDog	0.6816	0.7152	0.5816	0.8025	0.7741	0.6962	0.6968	0.7741
Average	0.7890	0.7721	0.7463	0.8180	0.8168	0.8008	0.7975	0.8140

The outer vertical lines show the entire range of values for the distribution. As shown in Fig. (**6**), the spread of the error distribution for SDP-LDA is narrow and close to 1, followed by S-LDA and C-LDA. The results on both the image and binary data sets clearly demonstrate that SDP-LDA obtained the most robust performance over these data sets.

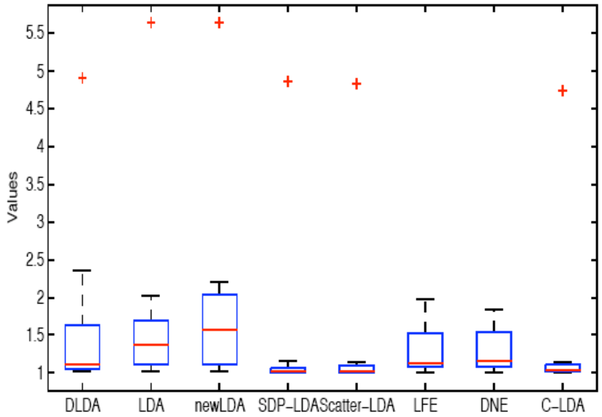

Fig. (6). Error distributions of DLDA, PCA+LDA, newLDA, SDP-LDA, S-LDA, LFE, DNE and C-LDA on the 13 data sets.

8. SUMMARY

This paper presents a margin based criterion for dimensionality reduction that potentially provides a solution to the small sample size problem, often associated with the Fisher criterion. In particular, the paper has shown that (1) the proposed criterion (15) for dimensionality deduction is closely related to the average margin criterion; (2) the criterion does not involve the inverse of S_w and can be optimized using algorithms such as semi-definite programming, thereby avoiding the small sample size problem; and (3) an error bound is established for the proposed technique. The paper demonstrates the efficacy of the proposed technique using a number of real examples, and the results show that the proposed technique registered competitive performance against several competing methods in several examples.

A. Proof of Lemma 3

Proof. We rewrite Eq. (19) as

$$\sum_{i=1}^{l}(y_i - w^t x_i)^2 + \gamma l w^t w = (y - X^t w)^t (y - X^t w) + \gamma l w^t w \qquad (29)$$

$$= l - 2(l_1 m_1 - l_2 m_2)^t w + w^t (S_m + \gamma lI) w \qquad (30)$$

where $X = (x_1 x_2 \cdots x_l)$ and $y = (y_1 y_2 \cdots y_l)^t$. Here we use the fact that $Xy = l_1 m_1 - l_2 m_2$. Taking the derivative with respect to w and setting the result to 0, we have

$$(S_m + \gamma lI) w = (l_1 m_1 - l_2 m_2). \qquad (31)$$

We show that three equations $(S_m + \gamma lI) w = (l_1 m_1 - l_2 m_2)$, $(S_m + \gamma lI) w = m_1 - m_2$, and $(S_w + \gamma lI) w = m_1 - m_2$ have the same solution w up to a constant, given that the overall mean is 0.

First we show that two equations $(S_m + \gamma lI) w = m$ and $(S_w + \gamma lI) w = m$ have the same solution (same set of eigenvectors), where $m = m_1 - m_2$ is the mean difference of the two classes.

Clearly solving $S_w w = m$ is equivalent to solving [2]

$$(S_w + \gamma lI)^{-1} S_b \Phi = \Phi \Lambda \qquad (32)$$

where Φ and Λ are the eigenvector and eigenvalue matrices of $(S_w + \gamma lI)^{-1} S_b$. Since we have $S_w = S_m - S_b$, following [2] (pp. 454), Eq. (32) can be written as

$$(S_m - S_b + \gamma lI)\Phi\Lambda = S_b\Phi$$

$$(S_m + \gamma lI)\Phi\Lambda = S_b\Phi(I + \Lambda)$$

$$(S_m + \gamma lI)^{-1} S_b\Phi = \Phi\Lambda(I + \Lambda)^{-1}. \tag{33}$$

This shows that Φ is also the eigenvector matrix of $(S_m + \gamma lI)^{-1} S_b$, and its eigenvalue matrix is $\Lambda(I + \Lambda)^{-1}$.

Without loss of generality, let us assume that the components α_i of Λ are such that $\alpha_1 \geq \cdots \geq \alpha_n$. It can be shown that the corresponding components of $\Lambda(I + \Lambda)^{-1}$ preserve the same relationship

$$\frac{\alpha_1}{1+\alpha_1} \geq \cdots \geq \frac{\alpha_n}{1+\alpha_n}.$$

That is, the t eigenvectors of $(S_m + \gamma lI)^{-1} S_b$ corresponding to the t largest eigenvalues are the same as the first t eigenvectors of matrix $(S_w + \gamma lI)^{-1} S_b$. As a special case (two class problem), the solutions resulting from $(S_m + \gamma lI)w = m$ and $(S_w + \gamma lI)w = m$ share the same "eigenvector".

Now we show that $(S_m + \gamma lI)w = (l_1 m_1 - l_2 m_2)$ and $(S_m + \gamma lI)w = m_1 - m_2$ produce the same solution as well. Consider that the overall mean m_0 is 0. From $lm_0 = l_1 m_1 + l_2 m_2 = 0$, we have $m_1 = \frac{l_2}{l} m, m_2 = -\frac{l_1}{l} m$, and $l_1 m_1 - l_2 m_2 = \frac{2l_1 l_2}{l} m.$ Thus

$$(S_m + \gamma lI)w = (l_1 m_1 - l_2 m_2)$$

becomes

$$(S_m + \gamma lI)w = \frac{2l_1 l_2}{l} m.$$

With constant $c = \frac{2l_1 l_2}{l}$, the solution of $(S_m + \gamma lI)w = cm$ is still in the same direction along the mean difference m, and thus is equivalent to solving $(S_w + \gamma lI)^{-1} S_b\Phi = \Phi\Lambda$.

REFERENCES

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Regression Trees*. Monterey, CA: Wadsworth and Brooks, 1984.

[2] K. Fukunaga, *Introduction to statistical pattern recognition*. Academic Press, 1990, 23.

[3] J. H. Friedman, "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165-175, 1989.

[4] P. Bickel and E. Levina, "Some theory for fisher's linear discriminant function, naive bayes and some alternatives when there are many more variables than observations," *Bernoulli*, vol. 10, no. 6, pp. 989-1010, 2004.

[5] H. Li, T. Jiang, and K. Zhang, "Efficient and robust feature extraction by maximum margin criterion," *IEEE Transactions on Neural Networks*, vol. 17, no. 1, pp. 157-165, 2006.

[6] J. Peng and S. Robila, "Weighted additive criterion for linear dimension reduction," in *Proceedings of IEEE International Conference on Data Mining*, 2007, pp. 619-624.

[7] V. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, 1997.

[8] L. F. Chen, H. Y. Liao, M. T. Ko, J. C. Lin, and G. J. Yu, "A new lda-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, vol. 33, pp. 1713-1726, 2001.

[9] R. Huang, Q. Liu, H. Lu, and S. Ma, "Solving the small sample size problem of lda," in *Proceedings of 16th International Conference on Pattern Recognition*, vol. 3, pp. 29-32, 2002.

[10] H. Yu and J. Yang, "A direct lda algorithm for high-dimension data with application to face recognition," *Pattern Recognition*, vol. 34, pp. 2067-2070, 2001.

[11] Y. Sun and D. Wu, "A relief based feature extraction algorithm," in *SIAM International Conference on Data Mining*, 2008, pp. 188-195.

[12] R. Gilad-Bachrach, A. Navot, and N. Tishby, "Margin based feature selection - theory and algorithms," in *ICML*, 2004.

[13] W. Zhang, X. Xue, Z. Sun, Y. Guo, and H. Lu, "Optimal dimensionality of metric space for classification," in *ICML*, 2007, 24.

[14] P. Zhang, J. Peng, and c. Domeniconi, "Kernel pooled local subspaces for classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 35, no. 3, pp. 489-502, 2005.

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

[16] L. Rueda and M. Herrera, "Linear dimensionality reduction by maximizing the chernoff distance in the transformed space," *Pattern Recognition*, vol. 41, no. 10, pp. 3138-3152, 2008.

[17] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Review*, vol. 38, no. 1, pp. 49-95, 1996.

[18] A. Ben-Tal and A. Nemirovski, "Non-euclidean restricted memory level method for large-scale convex optimization," *Mathematical Programming series A and B*, vol. 102, pp. 407-456, 2005.

[19] I. Nesterov, "Smooth minimization of non-smooth functions," 2003.

[20] H. Tang, T. Fang, and P. Shi, "Laplacian linear discriminant analysis," *Pattern Recognition*, vol. 39, pp. 136-139, 2006.

[21] F. Cucker and S. Smale, "Best choices for regularization parameters in learning theory: On the biasvariance problem," *Foundations of Computational Mathematics*, vol. 2, no. 4, pp. 413-428, 2002.

[22] T. Poggio and S. Smale, "The mathematics of learning: Dealing with data," *Notices of the American Mathematical Society*, vol. 50, no. 5, pp. 537-544, 2003.

[23] S. Smale and D. X. Zhou, "Shannon sampling II: Connection to learning theory," *Preprint*, pp. 1-21, 2004.

[24] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 12, pp. 625-653, 1999.

[25] D. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831-836, 1996, 25.

[26] K. Liu, Y. Cheng, and J. Yang, "A generalized optimal set of discriminant vectors," *Pattern Recognition*, vol. 25, no. 7, pp. 731-739, 1992.

[27] P. Phillips, "The facial recognition technology (feret) database," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, 2004.

[28] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, 1994, pp. 138-142.

Received: March 17, 2010 Revised: May 18, 2010 Accepted: May 26, 2010