Marginal Based Dimensionality Reduction and Generalization

Jing Peng∗,1, Stefan Robila1, Wei Fan2 and Guna Seetharaman3

1Computer Science Department, Montclair State University, Montclair, NJ 07043, USA
2IBM T.J. Watson Research, Hawthorne, NY 10532, USA
3Computing 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 discriminant 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 paper, we are concerned with LDA.

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

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

These examples are independently and identically distributed (i.i.d.) from the probability space $Z = X \times Y.$

Here probability measure $\rho$ is defined but unknown, $x_i \in \mathbb{R}^q$ are the q-dimensional inputs, and $y_i \in \mathbb{Y} = \{-M, M\} \subset \mathbb{R}$ are scalar labels. According to Fisher's criterion, one has to find a projection matrix $W \in \mathbb{R}^{r \times d}$ that maximizes:

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

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 $\varepsilon I$ is added, where $\varepsilon$ is a small positive number and $I$ the identity matrix of proper size. This results in maximizing

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

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
Margin Based Dimensionality Reduction and Generalization

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.
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 ($S_u$ becomes singular). A straightforward method is to use the pseudo-inverse of $S_u^+$ in place of $S_u^{-1}$. While simple, the method does not guarantee that Fisher's objective will be optimized by the eigenvector matrix of $S_u^+ S_u$. Furthermore, computing $S_u^+$ itself is ill posed. Another simple method is to first use PCA to remove the null space of $S_u$, 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 $S_u$ potentially contains discriminant information [8].

Another technique, newLDA [8], first transforms the data into the null space of $S_u$. 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 $S_u + S_b$, that is the common null space of both $S_u$ 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 $S_u + S_b$ replaces $S_u$, 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]

$$\sigma(x) = \| x - nm(x) \| - \| x - nh(x) \|.$$  \hspace{1cm} (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.

$$S_h = \sum_{i=1}^l h(x_i) h(x_i)'$$ \hspace{1cm} (5)

and

$$S_m = \sum_{i=1}^l m(x_i) m(x_i)'.$$ \hspace{1cm} (6)

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

$$\max_x x'(S_m - S_h)x$$ \hspace{1cm} (7)

$$(x' x)^{1/2} = 1$$

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

$$S_h = \sum_{x \in \{+1\}}(x - m_{i+1})(x - m_{i+1})' + \sum_{x \in \{-1\}}(x - m_{i-1})(x - m_{i-1})'$$ \hspace{1cm} (8)

$$= \sum_{x \in \{+1\}}xx' - 2 \sum_{x \in \{+1\}}xm_{i+1}' + \sum m_{i+1}m_{i+1}' + 2 \sum m_{i+1}m_{i-1}'$$

and

$$S_m = \sum_{x \in \{-1\}} xx' = \frac{1}{2} m_{i+1}m_{i+1}' - \frac{1}{2} m_{i-1}m_{i-1}'$$

Similarly, $S_m$ can be written as

$$S_m = \sum_{x \in \{+1\}}(x - m_{i+1})(x - m_{i+1})' + \sum_{x \in \{-1\}}(x - m_{i-1})(x - m_{i-1})'$$ \hspace{1cm} (9)

$$= \sum_{x \in \{+1\}} xx' - lm_{i+1}m_{i+1}' - lm_{i-1}m_{i-1}' + \frac{1}{2} m_{i+1}m_{i+1}' + \frac{1}{2} m_{i-1}m_{i-1}'$$

Then

$$S_m - S_h = l(m_{i+1}m_{i+1}' - 2m_{i+1}m_{i-1}' + m_{i-1}m_{i-1}')$$ \hspace{1cm} (10)

$$= lS_b.$$
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
\( x_j \in \text{NB}_u(x_i) \) if \( x_j \) is a within class neighbor of \( x_i \), and
\( x_j \in \text{NB}_b(x_i) \) 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
\[
F_{ij} = \begin{cases} 
1 & \text{if } x_i \in \text{NB}_u(x_j) \text{ or } x_j \in \text{NB}_u(x_i); \\
-1 & \text{if } x_i \in \text{NB}_b(x_j) \text{ or } x_j \in \text{NB}_b(x_i); \\
0 & \text{otherwise}.
\end{cases}
\]

The objective is to find \( P \) such that
\[
\text{tr}(P'X(S - F)X'P)
\]
is minimized, subject to \( P'P = I \). Here \( X \) is the data matrix,
and \( S \) is a diagonal matrix, where \( S_u = \sum_j F_{ij} \) or
\( S_b = \sum_j F_{ij} \).

If we use \( k = 1 \) to compute \( \text{NB}_u \) and \( \text{NB}_b \), we can write
\( X(S - F) \) as
\[
M = (mh(x_1) \cdots mh(x_i)),
\]
where \( mh(x_i) = mm(x_i) - nh(x_i) \) represents the difference
between the near miss \( mm(x_i) \) and the near hit \( nh(x_i) \) of \( x_i \),
respectively [12]. In this case, the objective becomes maximizing
\( \text{tr}(P'MX'P) \) over \( P \), subject to \( P'P = I \). This is in many ways
closest to the idea presented in [14] where we compute a
subspace by pooling local information
\( (mm(x_i) - nh(x_i))X_i' \), for \( i = 1, \ldots, l \). Here the local
information is the cross covariance of an instance and the
difference between its near miss and near hit. If we compute
\( \text{NB}_u \) and \( \text{NB}_b \) over the entire classes, i.e., \( k = l/2 \)
(assuming each class has the same number of examples), it
can be shown that
\[
X(S - F)X' = S_b.
\]
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 \( m_n \) 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
\[
S_w = \sum_j p_j (x_j - m_j)(x_j - m_j)' ,
\]
where \( p_j \) 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
\[
S_b = \sum_j p_j (m_j - m_0)(m_j - m_0)' .
\]
The mixture scatter matrix is the covariance matrix of all
samples, regardless of their class assignment, and it is given
by
\[
S_w = \sum_i (x_i - m_0)(x_i - m_0)' = S_u + S_b.
\]
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: \( S_bw = \lambda S_ww \). The eigenvectors
are the largest eigenvalues form the columns of \( W \). For a
two class problem, it can be written in a simpler form:
\( S_ww = m = m_1 - m_2 \), 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
\[
J(w) = \text{tr}(w' (\lambda S_b - S_u)w),
\]
where \( \text{tr} \) denotes the trace operator, and \( \lambda > 0 \) is a constant
that weights relative importance of the two terms \( S_b \) and \( S_u \)
in determining the outcome of linear discriminants. Large \( \lambda \)
values ignore within class spread, while small \( \lambda \) values
penalize discriminants that result in large within class variations.

Notice that \( tr(S_c) \) measures the overall scatter of class means. Therefore, a large \( tr(S_c) \) implies that the class means spread out in a transformed space. On the other hand, a small \( tr(S_c) \) 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 \( tr(S_c - S_m) \) is equivalent to maximizing

\[
J = \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{q} p_i p_j d(C_i, C_j),
\]

where \( p_i \) denotes the probability of class \( C_i \). The interclass distance \( d \) is defined as

\[
d(C_i, C_j) = d(m_i, m_j) - tr(S_c) = tr(S_c),
\]

where \( m_i \) represents the mean of class \( C_i \), and \( S_c \) 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 \( S_c \), thereby avoiding the small sample size problem associated with the Fisher criterion.

4. COMPUTING LINEAR DISCRIMINANTS WITH SEMIDEFINITE 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

\[
\max \quad tr(w'(\lambda S_c - S_m)w)
\]

subject to: \( \| w \| = 1 \).

This is a constraint optimization problem. Since \( tr(w'(\lambda S_c - S_m)w) = tr((\lambda S_c - S_m)ww') = tr((\lambda S_c - S_m)X) \), where \( X =ww' \), we can rewrite the above constraint optimization problem as

\[
tr((\lambda S_c - S_m)X)
\]

\[I \bullet X = 1\]

\[X \geq 0\]  \hspace{1cm} (15)

where \( I \) is the identity matrix and the inner product of symmetric matrices is \( A \bullet B = \sum_{i,j} a_{ij} b_{ij} \), 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 \( \text{rank}(X) = 1 \).

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 SDPs, 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 \( \text{rank}(X) = 1 \). Since \( X \) is symmetric, one can show that \( \text{rank}(X) = 1 \) iff \( X = ww' \) 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

\[
w = X(:,i)/X(1,i),
\]

(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 \( \text{rank}(X) = 1 \) 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., \( \text{rank}(X) = 1 \).

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

Proof. We rewrite \( \lambda S_c - S_m = 2\lambda S_b - S_m \), where \( S_m = S_b + S_u \). Let \( \text{null}(A) \) denote the null space of matrix \( A \). Since \( \text{null}(S_m) \subseteq \text{null}(S_b) \), there exists a matrix \( P \in \mathbb{R}^{r \times s} \) 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 \( \Lambda_m \) 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 \( 2\lambda \Lambda_m - I \). We then have
$P^t S_b P = \Lambda_b$, \hspace{2em} $P^t S_w P = I$. \hspace{2em} (17)

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

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

$$
\max_{w} \text{tr}(W^t (2\lambda S_b - S_w) W) = \max_{y} \text{tr}((PY)^t (2\lambda S_b - S_w) PY),
$$
(2 \lambda S_b - S_w) PY = \max_{y} \text{tr}(Y^t (2 \lambda A - I) Y).

It is straightforward to show that the maximum is attained by $Y = [e_1 e_2 \cdots e_r; 0]$, 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 $X = \sum_{i=1}^r w_i w_i^t$. Thus,

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

However, the constraint $I \cdot X = 1$ states that $tr(X) = 1$. It follows that $r = 1$. That is, \text{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. 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 $\lambda S - b - S_e$, 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

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

when the hypothesis space $\mathcal{H}$ is linear $\mathcal{H}_L = \{ f \mid f(x) = w^t x \}$. Here $\gamma$ represents the regularization constant. In this case, we have $\| f \|_{H} = w^t w$, and thus

$$
w_{\text{opt}} = \arg \min_{w} \frac{1}{l} \sum_{i=1}^l (y_i - w^t x_i)^2 + \gamma w^t w.
$$
(19)

We then use the bound for $w_{\text{opt}}$ 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

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

where $\alpha$ 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

$$
(S_e + \alpha I) w = \lambda S_b w.
$$
(21)

Solving the above is equivalent to solving

$$
(S_e + \alpha I) w = m_1 - m_2,
$$
(22)

since $S_e w$ 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 $\epsilon$ in Eq. (3) to $\alpha$ 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 $w_{\text{opt}}$ 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 $w_{\text{opt}}$ 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 $f_p$ 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).
\[ f_\rho = \arg \min_{f} \int_X (y - f(x))^2 \, d\rho. \]  

(23)

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

\[ f_{opt} = \arg \min_{f} \int_X (f - f_{opt})^2 \, d\rho. \]

Here \( \rho \) is unknown and so is \( f_\rho \). 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 \( f_\rho \). 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

\[ L_k f(x) = \int f(x) k(x,x') \, d\rho_x, \]

(24)

where \( \rho_x \) 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

\[ |f(x) - y| \leq M \]

(25)

almost everywhere.

**Theorem 5** Let \( z \) Eq. (1) be randomly drawn according to \( \rho \), and \( f_\rho \) 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:

\[ \int (w_{opt} - f_{opt})^2 \, d\rho_x \leq S(\gamma) + A(\gamma), \]

(26)

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

\[ A(\gamma) = \gamma^{1/2} \left\| L_k^{1/2} f_{opt} \right\|^2 \]

and

\[ S(\gamma) = \frac{32M^2(\gamma + C_k)^2}{\gamma^2} \nu'(m, \delta), \]

where \( \nu'(m, \delta) \) is the unique positive solution of

\[ \frac{m}{4} \nu^3 - \ln(4m/\delta) \nu - c = 0. \]

(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 \( f_\rho \). \( S(\gamma) \) 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

\[ \gamma = \left( \frac{\log(4/\delta)^2 \frac{1}{\gamma}}{m} \right)^{1/2}. \]

(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 \( \gamma \). Eq. (28).

It shows that for a fixed \( \delta \), \( \gamma \) goes to zero as the number of training examples goes to infinity, as expected. For a given training data set, high confidence (low \( \delta \) ) requires large \( \gamma \). Notice that there is no conflict between \( \gamma \) defined in [21] and the one in [23]. \( \gamma \) 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 \( A_i = A S_b - S_w \), where \( S_b \) and \( S_w \) are computed as in the two class case. We solve the problem in (15)

\[ \max_X \quad \text{tr}(A_1 X) \]

\[ X \succeq 0 \]

\[ I \cdot X = 1 \]

to obtain the solution \( X_1 = w_j w_j' \). Once we have obtained the solution \( X_1 = w_j w_j' \), we inflate \( A_j \) to obtain

\[ A_{j+1} = A_j + X_j, \]

from which we compute \( X_{j+1} = \sum_{j=1}^{C-1} w_j w_j' \). Here \( X_j \) is an approximation to \( w_j \) for \( j = 1, \ldots, C-1 \), where \( C \) represents the number of classes. Here \( X_j \) s force \( w_{j+1} \) to be orthogonal to \( w_j \), as desired. To see this, we write

\[ w_{j+1}(A_j) w_{j+1} \]

\[ = w_{j+1} (A_1 + w_i w_i' + \cdots + w_j w_j') w_{j+1} \]

\[ = w_{j+1} A_{j+1} + (w_{j+1} w_j')^2 + \cdots + (w_{j+1} w_j')^2. \]

Thus, \( w_{j+1}(A_j) w_{j+1} \) is minimized when \( w_{j+1} \) minimizes \( A_j \) and is orthogonal to the \( w_j \) s, since \( w_j = 0 \) 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]:
\[
J(W) = |W^T S_b W|/|W^T S_w W|.
\]

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

**DLDA:** Apply PCA to remove the null space of \( S_w \) 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 calculated 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 \( 384 \times 256 \) 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 \( 150 \times 130 \) pixels. The corresponding preprocessed images are shown in Fig. (2). The preprocessed images of \( 150 \times 130 \) 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.

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
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.

![ORL sample images](image)

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.

![Comparison of SDP-LDA, DLDA, LDA, newLDA, S-LDA, and DNE on the ORL image data.](image)

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 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:

\[
b_m = \frac{e_m}{\min_{i \neq m} e_i}.
\]

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.
vides a solution—setting the result to 0, we have
is equivalent to solving
and can be up to a constant, given that the overall
competing methods
and
and
\( m \) can be generated.
Proof. We rewrite Eq. (19) as
\[
\sum_{i=1}^{l} (y_i - w^t x_i)^2 + \gamma l w^t w = (y - X^t w)'(y - X^t w) + \gamma l w^t w
\]
\[
= l - 2(l_1 m_1 - l_2 m_2)' w + w'(S_m + \gamma l l) w
\]
where \( X = (x_1, x_2, \ldots, x_i) \) and \( y = (y_1, y_2, \ldots, y_l)' \). Here we use the fact that \( X y = l_1 m_1 - l_2 m_2 \). Taking the derivative with respect to \( w \) and setting the result to 0, we have
\[
(S_m + \gamma l l) w = (l_1 m_1 - l_2 m_2).
\]
We show that three equations
(\( S_m + \gamma l l) w = (l_1 m_1 - l_2 m_2) \),
(\( S_m + \gamma l l) w = m_1 - m_2 \), and (\( S_m + \gamma l l) w = m_1 - m_2 \) have the same solution \( w \) up to a constant, given that the overall mean is 0.
First we show that two equations
(\( S_m + \gamma l l) w = m \) and
(\( S_m + \gamma l l) w = m \) have the same solution (same set of eigenvectors), where \( m = m_1 - m_2 \) is the mean difference of the two classes.
 Clearly solving \( S_m w = m \) is equivalent to solving
(\( S_m + \gamma l l \) is a positive definite matrix).

### 8. SUMMARY

This paper presents a margin based criterion for dimensionality reduction that potentially provides a solution to 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
\[
\sum_{i=1}^{l} (y_i - w^t x_i)^2 + \gamma l w^t w = (y - X^t w)'(y - X^t w) + \gamma l w^t w
\]
\[
= l - 2(l_1 m_1 - l_2 m_2)' w + w'(S_m + \gamma l l) w
\]
where \( X = (x_1, x_2, \ldots, x_i) \) and \( y = (y_1, y_2, \ldots, y_l)' \). Here we use the fact that \( X y = l_1 m_1 - l_2 m_2 \). Taking the derivative with respect to \( w \) and setting the result to 0, we have
\[
(S_m + \gamma l l) w = (l_1 m_1 - l_2 m_2).
\]
We show that three equations
(\( S_m + \gamma l l) w = (l_1 m_1 - l_2 m_2) \),
(\( S_m + \gamma l l) w = m_1 - m_2 \), and (\( S_m + \gamma l l) w = m_1 - m_2 \) have the same solution \( w \) up to a constant, given that the overall mean is 0.

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

Clearly solving \( S_m w = m \) is equivalent to solving
(\( S_m + \gamma l l \) is a positive definite matrix).
where $\Phi$ and $\Lambda$ are the eigenvector and eigenvalue matrices of $(S_m + \gamma I)^{-1} S_b$. Since we have $S_y = S_m - S_b$, following [2] (pp. 454), Eq. (32) can be written as

\[(S_m - S_y + \gamma I)\Phi\Lambda = S_y\Phi \]

\[(S_m + \gamma I)\Phi\Lambda = S_y(\Phi(I + \Lambda))^{-1}. \]

(33)

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

Without loss of generality, let us assume that the components $\alpha_i$ of $\Lambda$ 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 $(S_m + \gamma I)^{-1} S_b$ corresponding to the $t$ largest eigenvalues are the same as the first $t$ eigenvectors of matrix $(S_m + \gamma I)^{-1} S_y$. As a special case (two class problem), the solutions resulting from $(S_m + \gamma I)w = m$ and $(S_m + \gamma I)w = m$ share the same “eigenvector”.

Now we show that $(S_m + \gamma I)w = (l_1 m_1 - l_2 m_2)$ and $(S_m + \gamma I)w = m_1 - m_2$ produce the same solution as well. Consider that the overall mean $m_0 = 0$. From

\[l_m = l_1 m_1 + l_2 m_2 = 0, \quad m = l_1 m_1 l_2 m_2 = \frac{l_1(l_2 - l_1)}{l_2} m, \quad \text{and} \]

\[l_1 m_1 - l_2 m_2 = \frac{2l_1 l_2}{l_2} m. \]

This becomes

\[(S_m + \gamma I)w = \frac{2l_1 l_2}{l_2} m. \]

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

REFERENCES


