Regularized Discriminant Analysis for High Dimensional, Low Sample Size Data

Jieping Ye
Arizona State University
Tempe, AZ 85287
jieping.ye@asu.edu

Tie Wang
Arizona State University
Tempe, AZ 85287
tie.wang@asu.edu

ABSTRACT

Linear and Quadratic Discriminant Analysis have been used widely in many areas of data mining, machine learning, and bioinformatics. Friedman proposed a compromise between Linear and Quadratic Discriminant Analysis, called Regularized Discriminant Analysis (RDA), which has been shown to be more flexible in dealing with various class distributions. RDA applies the regularization techniques by employing two regularization parameters, which are chosen to jointly maximize the classification performance. The optimal pair of parameters is commonly estimated via cross-validation from a set of candidate pairs. It is computationally prohibitive for high dimensional data, especially when the candidate set is large, which limits the applications of RDA to low dimensional data.

In this paper, a novel algorithm for RDA is presented for high dimensional data. It can estimate the optimal regularization parameters from a large set of parameter candidates efficiently. Experiments on a variety of datasets confirm the claimed theoretical estimate of the efficiency, and also show that, for a properly chosen pair of regularization parameters, RDA performs favorably in classification, in comparison with other existing classification methods.

Categories and Subject Descriptors: H.2.8 [Database Management]: Database Applications - Data Mining

General Terms: Algorithms

Keywords: Dimensionality reduction, Quadratic Discriminant Analysis, regularization, cross-validation

1. INTRODUCTION

Statistical discriminant analysis is a frequently used and widely applicable tool in a variety of areas [6, 12, 26, 27]. The aim of discriminant analysis is to assign a data point to one of several classes (groups) on the basis of a number of feature variables. Numerous methods on discriminant analysis have been proposed and applied in the past. The most frequently used methods are parametric approaches, especially Linear and Quadratic Discriminant Analysis. Linear Discriminant Analysis (LDA) is based on the assumption that the variables are multivariate normally distributed in each class with different mean vectors and a common covariance matrix. It has been used in various applications [1, 6, 19, 24]. In Quadratic Discriminant Analysis (QDA), the variables are assumed to be multivariate normally distributed in each class with different mean vectors and different covariance matrices [14]. QDA provides a less restrictive procedure by allowing different covariance matrices and may thus fit the data better than LDA. However, LDA involves a much smaller number of parameters to estimate than QDA, and is thus more robust and reliable than QDA in the parameter estimation.

Friedman [8] proposed a compromise between LDA and QDA, called Regularized Discriminant Analysis (or RDA in short), which allows one to shrink the separate covariances of QDA toward a common covariance as in LDA. The regularized covariance matrix of the i-th class has the following form:

$$\beta\left(\alpha\Sigma_i + (1-\alpha)S_w\right) + (1-\beta)I_d,$$

where Σ_i is the covariance of the i-th class, S_w, the so-called pooled covariance matrix as used in LDA, is also known as the within-class scatter matrix [9], I_d is the identity matrix of size d by d, and d is the dimensionality of the data. Here $\alpha \in [0,1]$ and $\beta \in [0,1]$ are two regularization parameters. The trace term in the original RDA formulation in [8] is absorbed into the β parameter for simplicity.

RDA provides a fairly rich class of regularization alternatives. The four corners defining the extremes of the (α, β) plane represent well-known classification procedures. The upper right corner ($\alpha = 1, \beta = 1$) represents QDA. The upper left corner ($\alpha = 0, \beta = 1$) represents LDA. The line connecting the lower left and lower right corner, i.e., $\beta = 0$ with $0 \leq \alpha \leq 1$, corresponds to the nearest-centroid classifier well known in pattern recognition, where a test data point is assigned to the class with the closest (Euclidean distance) centroid. Varying α with β fixed at 1 produces models between QDA and LDA. For a given training dataset, α and β are commonly estimated via cross-validation. Selecting an optimal value for a parameter pair such as (α, β) is called *model selection* [14]. The computational cost of model selection for RDA is high, especially when the data dimensionality, d, is large, since it requires expensive matrix computations for each candidate pair. This restricts RDA to low dimensional data.

In this paper, we make the first attempt in extending

KDD'06, August 20–23, 2006, Philadelphia, Pennsylvania, USA.

the applicability of RDA to high dimensional, low sample size (HDLSS) data, as HDLSS data are emerging from various fields. In high throughput gene expression experiments, technologies have been designed to measure the gene expression levels of tens of thousands of genes in a single microarray chip. However, the sample size in each dataset is typically small ranging from tens to low hundreds due to the cost of the experiments. In image-based object or face recognition applications, two or three dimensional images are usually converted to column representations, resulting in high dimensional data while the number of images is usually small. In text document classification, the number of features equals to the number of distinct words in the documents, which is typically in the thousands, while the number of documents in the study may be much smaller. A common characteristic of all these datasets is that the dimensionality, d, of the data vector is much larger than the sample size. This leads to various statistical issues, known as the high dimensional, low sample size problem [13]. In this paper, we propose an efficient algorithm for RDA on HDLSS data. The primary contributions of this work include:

- We show that the classification rule in RDA can be decomposed into two components: the first component involves matrices of low dimensionality, while the second component involves matrices of high dimensionality. More importantly, we show that for a given test data point, the second component in the classification rule is constant for all classes, which has no effect on classification and can be simply removed. We call this the *decomposition property* of RDA.
- We present an efficient algorithm for RDA, by applying the decomposition property above to speed up the model selection process of RDA. The basic idea is to divide the computations in RDA into two successive stages. The first stage has a relatively higher computational cost, but it is independent of α and β. The second stage has a relatively lower computational cost. When searching for the optimal parameter pair from a set of candidates via cross-validation, we only need to repeat the second stage, thus dramatically reducing the computational cost of model selection, especially when the candidate set is large.
- We have conducted experimental studies on a variety of HDLSS data, including text documents, face images, and microarray gene expression data. Results confirm our theoretical estimate of the computational cost of the proposed RDA algorithm in model selection. Experiments also demonstrate that, with properly chosen regularization parameters, RDA is effective in classification, in comparison with several other known classification algorithms.

The rest of the paper is organized as follows. An overview of QDA and RDA is given in Section 2. An efficient algorithm for RDA is presented in Section 3. Experimental results are given in Section 4. Conclusions are presented in Section 5.

2. AN OVERVIEW OF QDA AND RDA

For convenience, we present in Table 1 the important notations that will be used in the rest of the paper.

Notation	Description
n	sample size
d	number of features (dimensions)
k	number of classes
A	data matrix
A_i	data matrix of the i-th class
n_i	size of the i-th class
μ_i	centroid of the i-th class
Σ_i	covariance matrix of the i-th class
$\hat{\Sigma}_i$	regularized covariance matrix of the i-th class
μ	global centroid of the training set
S_b	between-class scatter matrix
S_w	within-class scatter matrix
S_t	total scatter matrix
t	rank of the matrix S_t
α	the first regularization parameter
β	the second regularization parameter
$g(x)$	class label of the data point x

Table 1: Important notations used in the paper.

In this section, we briefly review the Quadratic Discriminant Analysis (QDA), some issues related to the application of QDA on HDLSS data, and the Regularized Discriminant Analysis (RDA). Note that LDA is a special case of QDA when all classes share a common class covariance.

Given a training dataset of n data points $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ is the feature vector of the i-th data point, d is the data dimensionality, $y_i = g(x_i) \in \{1, 2, \cdots, k\}$ is the class label of x_i, and k is the number of classes. Let

$$A = [x_1, x_2, \cdots, x_n] \in \mathbb{R}^{d \times n}$$

be the data matrix, which can be decomposed into k classes as $A = [A_1, A_2, \cdots, A_k]$, where A_i contains all data points from the i-th class. Denote $n_i = |A_i|$ as the size of the i-th class. We have $n = \sum_{i=1}^k n_i$.

Assuming the class densities follow the normal distribution, we apply the following classification rule [8, 14]: a test point $x \in \mathbb{R}^d$ is classified as class $C(x)$ defined by

$$C(x) = \text{argmin}_i \left\{ (x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i) + \ln|\Sigma_i| \right\}, \quad (1)$$

where the centroid μ_i of i-th class is defined as

$$\mu_i = \frac{1}{n_i} A_i e^{(i)}, \quad (2)$$

$e^{(i)} \in \mathbb{R}^{n_i}$ is a vector of all ones, and the covariance matrix Σ_i of i-th class is defined as

$$\Sigma_i = \frac{1}{n_i} \sum_{x \in A_i} (x - \mu_i)(x - \mu_i)^T. \quad (3)$$

Note that we have assumed an equal prior for all classes in Eq. (1) for simplicity.

The decision boundary using the above classification rule is quadratic and the algorithm is thus called Quadratic Discriminant Analysis (QDA). In a special case where all classes share a common covariance, that is, $\Sigma_i = \Sigma_j$, for any class i and class j, QDA is reduced to the well-known Linear Discriminant Analysis (LDA) [5, 7, 9, 14].

The traditional QDA formulation in Eq. (1) requires all class covariance matrices to be nonsingular. However, for

many applications involving HDLSS data, such as text document classification, face recognition, and microarray gene expression data analysis, all class covariance matrices may be singular, since the data dimensionality may be much larger than the sample size of all classes in the training dataset. Furthermore, the estimates of the class covariance matrices may be biased and unreliable. As pointed out in [8], this bias is more pronounced, when their eigenvalues tend toward equality, while it is correspondingly less severe when their eigenvalues are highly disparate. In both cases, this phenomenon becomes more pronounced as the sample size decreases. Thus, HDLSS data presents a major challenge for the application of QDA. In [8], Friedman proposed a compromise between LDA and QDA, called Regularized Discriminant Analysis (RDA), which allows one to shrink the separate covariances of QDA toward a common covariance as in LDA by employing regularization techniques. Regularization has been commonly used in the solution of ill-posed inverse problems [20], where the number of parameters exceeds the sample size. In such cases, the parameter estimates can be highly unstable, giving rise to high variance. By employing a method of regularization, one attempts to improve the estimates by regulating this bias variance trade-off.

Quadratic Discriminant Analysis is ill-posed if $n_k < d$ for any class. One method of regularization is to replace the individual class covariance matrix Σ_i by $S_i(\alpha)$ as follows:

$$S_i(\alpha) = \alpha\Sigma_i + (1-\alpha)S_w, \tag{4}$$

where S_w is the weighted average of the class covariance matrices, called pooled covariance matrix, or within-class scatter matrix [9], which is defined as

$$S_w = \frac{1}{n}\sum_{i=1}^{k}\sum_{x\in A_i}(x-\mu_i)(x-\mu_i)^T = \frac{1}{n}\sum_{i=1}^{k}(n_i\Sigma_i). \tag{5}$$

The regularization parameter α takes on values between 0 and 1. It controls the degree of shrinkage of the individual class covariance matrix estimates toward the pooled estimate. The value $\alpha = 1$ gives rise to QDA, whereas $\alpha = 0$ yields LDA.

However, the regularization in Eq. (4) is still fairly limited. First, it might not provide enough regularization. If the total sample size, n, is less than the data dimensionality, d, then even LDA is ill-posed [17, 22]. Second, biasing the class covariance matrices toward commonality may not be the most effective way to shrink them. Recall that ridge regression regularizes ordinary linear least squares regression by shrinking toward a multiple of the identity matrix [14, 15]. To this end, a further regularization is given by

$$\hat{\Sigma}_i = \beta S_i(\alpha) + (1-\beta)I_d, \tag{6}$$

where I_d is the identity matrix of d by d and β is an additional regularization parameter, which controls shrinkage toward a multiple of the identity matrix.

In this paper, we apply a variant of the regularized class covariance matrix in Eq. (6), given by

$$\hat{\Sigma}_i = \beta\left(\alpha\Sigma_i + (1-\alpha)S_t\right) + (1-\beta)I_d, \tag{7}$$

where total scatter matrix S_t is defined as

$$S_t = \frac{1}{n}\sum_{i=1}^{n}(x_i-\mu)(x_i-\mu)^T, \tag{8}$$

$\alpha \in [0,1]$, and $\beta \in [0,1]$. The minor difference here lies in the matrix S_t used in Eq. (7), while S_w is employed in Eq. (6). It is interesting to note that, when $\beta \to 1$, the classification rule based on the regularized class covariance matrix in Eq. (6) may be numerically unstable, while the one based on the matrix in Eq. (7) is stable even for HDLSS data (see Section 3). The use of S_t instead of S_w has recently been explored in LDA for improving numerical stability [3, 22].

A test point x in RDA is classified as class $\hat{C}(x)$ given by

$$\hat{C}(x) = \text{argmin}_i\left\{(x-\mu_i)^T\hat{\Sigma}_i^{-1}(x-\mu_i) + \ln|\hat{\Sigma}_i|\right\}, \tag{9}$$

where $\hat{\Sigma}_i$ is defined in Eq. (7). The performance of RDA may be critically dependent on the value of the parameters α and β. Cross-validation is commonly used to estimate the optimal α and β from a finite set, $\Lambda = \{(\alpha_i, \beta_j)\}$, where $i = 1, \cdots, r$, and $j = 1, \cdots, s$. The number of candidate pairs (α, β) is $|\Lambda| = rs$. In practice, a large number, rs, of candidate pairs is often desirable to achieve good classification performance. However, with a large number of parameter pairs, the computational cost of model selection for RDA may be prohibitive for HDLSS data, since it requires expensive matrix computations for each candidate pair.

A direct implementation of RDA as the one used in [8] involves the formation of $\hat{\Sigma}_i$ and the inversion of $\hat{\Sigma}_i$ for all i. The computation of the inversion of all k class covariance matrices takes $O(kd^3)$ time and is prohibitive for HDLSS data, where the data dimensionality d is large. This limits the applications of RDA to low dimensional data.

3. EFFICIENT MODEL SELECTION FOR RDA

In this section, we first establish a key property of RDA, which shows that the classification rule in RDA can be decomposed into two components. The first component involves matrices of low dimensionality, while the second component involves matrices of high dimensionality. More importantly, we show that for a given test data point, the second component in the classification rule is constant for all classes, which has no effect on the classification and can be simply removed. Thus, the computational cost of RDA can be significantly reduced. We call this the decomposition property of RDA.

We show below that the essence of the decomposition property of RDA is that the first component of the classification rule lies in the orthogonal complement of the null space of S_t, which has low dimensionality for HDLSS data, while the second component lies in the null space of S_t of high dimensionality.

Define the between-class scatter matrix S_b, used in discriminant analysis [9], as follows:

$$S_b = \frac{1}{n}\sum_{i=1}^{k} n_i(\mu_i-\mu)(\mu_i-\mu)^T. \tag{10}$$

It follows from the definition that

$$S_t = S_b + S_w. \tag{11}$$

We have the following result concerning the relationship between the null space of S_t and the null space of Σ_i and S_w:

LEMMA 3.1. *Let Σ_i, S_b, S_t, and S_w be defined as above. The null space of S_t denoted as $N(S_t)$ is a subset of the null space, $N(S_b)$ of S_b and a subset of the null space, $N(\Sigma_i)$ of Σ_i, for all i. That is, $N(S_t) \subseteq N(S_b)$ and $N(S_t) \subseteq N(\Sigma_i)$, for all i.*

PROOF. Consider any $x \in N(S_t)$. That is, $S_t x = 0$ and $x^T S_t x = 0$. From Eqs. (5) and (11), we have

$$S_t = \sum_{i=1}^{K} \frac{n_i}{n} \Sigma_i + S_b. \tag{12}$$

It follows that

$$\begin{aligned} 0 = x^T S_t x &= x^T \left(\sum_{i=1}^{K} \frac{n_i}{n} \Sigma_i + S_b \right) x \\ &= \sum_{i=1}^{K} \left(\frac{n_i}{n} x^T \Sigma_i x \right) + x^T S_b x. \end{aligned}$$

Since Σ_i, for all i, and S_b are positive semi-definite, we have $x^T \Sigma_i x = 0$, for all i, and $x^T S_b x = 0$. It follows that $\Sigma_i x = 0$ and $S_b x = 0$. Therefore, x also lies in the null space of Σ_i and S_b. Hence, $N(S_t) \subseteq N(S_b)$ and $N(S_t) \subseteq N(\Sigma_i)$. □

Let $S_t = UDU^T$ be the Singular Value Decomposition (SVD) [10] of S_t, where U is orthogonal, $D = \begin{pmatrix} D_t & 0 \\ 0 & 0 \end{pmatrix}$, $D_t \in \mathbb{R}^{t \times t}$ is diagonal and $t = \text{rank}(S_t)$. Note that $t \leq n$. Partition U into $U = [U_1, U_2]$, where $U_1 \in \mathbb{R}^{d \times t}$ and $U_2 \in \mathbb{R}^{d \times (d-t)}$. Then U_2 lies in the null space of S_t, i.e., $S_t U_2 = 0$. We have the following result concerning the decomposition structure of $\hat{\Sigma}_i$:

LEMMA 3.2. *Let $U = [U_1, U_2]$ be defined as above and let $\hat{\Sigma}_i$ be defined as in Eq. (7). Then $\hat{\Sigma}_i$ can be expressed as*

$$\hat{\Sigma}_i = U \begin{pmatrix} M_i & 0 \\ 0 & (1-\beta) I_{d-t} \end{pmatrix} U^T, \tag{13}$$

where

$$M_i = \beta \left(\alpha \tilde{\Sigma}_i + (1-\alpha) D_t \right) + (1-\beta) I_t, \tag{14}$$

and $\tilde{\Sigma}_i = U_1^T \Sigma_i U_1$.

PROOF. Recall from Eq. (7) that

$$\hat{\Sigma}_i = \beta \left(\alpha \Sigma_i + (1-\alpha) S_t \right) + (1-\beta) I_d.$$

It follows that

$$U^T \hat{\Sigma}_i U = \beta \left(\alpha U^T \Sigma_i U + (1-\alpha) U^T S_t U \right) + (1-\beta) I_d.$$

From Lemma 3.1, $\Sigma_i U_2 = 0$, for all i. It follows that

$$\begin{aligned} \hat{\Sigma}_i &= U \left(\beta \left(\alpha U^T \Sigma_i U + (1-\alpha) U^T S_t U \right) + (1-\beta) I_d \right) U^T \\ &= U \begin{pmatrix} M_i & 0 \\ 0 & (1-\beta) I_{d-t} \end{pmatrix} U^T. \end{aligned}$$

□

Lemma 3.2 implies that all k regularized class covariance matrices share a similar decomposition structure, which leads to the decomposition property of RDA as summarized in the following proposition:

PROPOSITION 3.1. *Let U_1, U_2, and M_i be defined as above. Then the classification rule in Eq. (9) is equivalent to:*

$$\begin{aligned} \hat{C}(x) = argmin_i \Big\{ &(x-\mu_i)^T U_1 M_i^{-1} U_1^T (x-\mu_i) + ln|M_i| \\ &+ (1-\beta)^{-1} (x-\mu_i)^T U_2 U_2^T (x-\mu_i) + (1-\beta)^{d-t} \Big\}. \end{aligned} \tag{15}$$

PROOF. Denote $F_i = (x-\mu_i)^T \hat{\Sigma}_i^{-1} (x-\mu_i)$. It follows from Lemma 3.2 that

$$\begin{aligned} F_i &= (x-\mu_i)^T \hat{\Sigma}_i^{-1} (x-\mu_i) \\ &= (x-\mu_i)^T U \begin{pmatrix} M_i & 0 \\ 0 & (1-\beta) I_{d-t} \end{pmatrix}^{-1} U^T (x-\mu_i) \\ &= (x-\mu_i)^T U_1 M_i^{-1} U_1^T (x-\mu_i) \\ &+ (1-\beta)^{-1} (x-\mu_i)^T U_2 U_2^T (x-\mu_i). \end{aligned} \tag{16}$$

The result follows directly from Lemma 3.2 and Eq. (16), as $\ln|\hat{\Sigma}_i| = \ln|M_i| + \ln|(1-\beta) I_{d-t}| = \ln|M_i| + (1-\beta)^{d-t}$. □

Proposition 3.1 implies that the classification rule in RDA can be decomposed into two components as in Eq. (15). The first component, i.e., $(x-\mu_i)^T U_1 M_i^{-1} U_1^T (x-\mu_i)$, involves U_1, which lies in the orthogonal complement of the null space of S_t, while the second component, i.e., $(1-\beta)^{-1}(x-\mu_i)^T U_2 U_2^T (x-\mu_i) + (1-\beta)^{d-t}$, involves U_2, which lies in the null space of S_t. Note that the null space of S_t is of dimension $d-t$, which is much larger than the dimension, t, of the orthogonal complement of the null space of S_t, for HDLSS data.

However, two issues need to be resolved before we apply the classification rule in Eq. (16). First, the computation may be numerically unstable as $\beta \to 1$, due to the presence of $(1-\beta)^{-1}$ in the computation. Second, finding the best parameter pair (α, β) from a set, Λ, of candidate pairs may be expensive, since $U_2 \in \mathbb{R}^{d \times (d-t)}$ is of large size for HDLSS data. Interestingly, both issues can be addressed simultaneously by simply removing the second term in Eq. (16), based on the lemma below:

LEMMA 3.3. *Let U_2, μ_i and μ be defined as above, then $U_2^T(\mu_i - \mu) = 0$. Thus, $U_2^T(x-\mu_i) = U_2^T(x-\mu)$, for any x.*

PROOF. Note that U_2 lies in the null space of S_t. From Lemma 3.1, U_2 also lies in the null space of S_b. That is, $U_2^T S_b = 0$. S_b in Eq. (10) can be expressed as $S_b = H_b H_b^T$, where

$$H_b = \frac{1}{\sqrt{n}} \left[\sqrt{n_1}(\mu_1 - \mu), \sqrt{n_2}(\mu_2 - \mu), \cdots, \sqrt{n_k}(\mu_k - \mu) \right]. \tag{17}$$

It follows from $U_2^T S_b = 0$ that $U_2^T H_b = 0$, i.e.,

$$U_2^T \left[\sqrt{n_1}(\mu_1 - \mu), \sqrt{n_2}(\mu_2 - \mu), \cdots, \sqrt{n_k}(\mu_k - \mu) \right] = 0,$$

and $U_2^T(\mu_i - \mu) = 0$. Hence, $U_2^T(x-\mu_i) = U_2^T(x-\mu)$. □

From Lemma 3.3, the classification rule in Eq. (15) can be further simplified by removing the second component as

$$\begin{aligned} \hat{C}(x) &= \text{argmin}_i \left\{ (x-\mu_i)^T U_1 M_i^{-1} U_1^T (x-\mu_i) + \ln|M_i| \right\} \\ &= \text{argmin}_i \left\{ (\tilde{x}-\tilde{\mu}_i)^T M_i^{-1} (\tilde{x}-\tilde{\mu}_i) + \ln|M_i| \right\}, \end{aligned} \tag{18}$$

where $\tilde{x} = U_1^T x$ and $\tilde{\mu}_i = U_1^T \mu_i$.

3.1 The computation of M_i^{-1} and $|M_i|$

The main computations in Eq. (18) are the inversion of M_i and the determinant of M_i, for all i, which take $O(kt^3) = O(kn^3)$ time, as $M_i \in \mathbb{R}^{t\times t}$ and $t \leq n$. Recall from Section 2 that the direct implementation of RDA computes the inversion of $\hat{\Sigma}_i$ for all i directly with the time complexity of $O(kd^3)$, which is significantly higher than $O(kn^3)$ for HDLSS data. In the following, we present an efficient way of computing the inversion of M_i and the determinant of M_i, for all i, with a time complexity of $O(n^3/k)$, thus further reducing the complexity of the algorithm.

Define the matrix $H_i \in \mathbb{R}^{d\times n_i}$ as follows:

$$H_i = \frac{1}{\sqrt{n_i}}[A_i - \mu_i(e^{(i)})^T], \tag{19}$$

where $A_i \in \mathbb{R}^{d\times n_i}$ is the data matrix of the i-th class, μ_i is the centroid of the i-th class, and $e^{(i)}$ is the vector of all ones of length n_i. Then the class covariance matrix, Σ_i, of the i-th class can be expressed as:

$$\Sigma_i = H_i H_i^T. \tag{20}$$

It follows that

$$\tilde{\Sigma}_i = U_1^T H_i H_i^T U_1 = \tilde{H}_i \tilde{H}_i^T, \tag{21}$$

where $\tilde{H}_i = U_1^T H_i$. Denote $D_{\alpha\beta}$ as the diagonal matrix:

$$D_{\alpha\beta} = (1-\alpha)\beta D_t + (1-\beta) I_t. \tag{22}$$

From Eq. (14), We have

$$\begin{aligned} M_i &= \alpha\beta \tilde{H}_i \tilde{H}_i^T + D_{\alpha\beta} \\ &= D_{\alpha\beta}^{0.5}\left((\sqrt{\alpha\beta} D_{\alpha\beta}^{-0.5}\tilde{H}_i)(\sqrt{\alpha\beta} D_{\alpha\beta}^{-0.5}\tilde{H}_i)^T + I_t\right) D_{\alpha\beta}^{0.5} \\ &= D_{\alpha\beta}^{0.5}\left(X_i X_i^T + I_t\right) D_{\alpha\beta}^{0.5}, \end{aligned} \tag{23}$$

where

$$X_i = \sqrt{\alpha\beta} D_{\alpha\beta}^{-0.5}\tilde{H}_i \in \mathbb{R}^{t\times n_i}.$$

It follows from the Sherman-Woodbury-Morrison formula [10] that

$$M_i^{-1} = D_{\alpha\beta}^{-0.5}\left(I_t - X_i(I_{n_i} + X_i^T X_i)^{-1} X_i^T\right) D_{\alpha\beta}^{-0.5}. \tag{24}$$

Note that the matrix inversion in Eq. (24) is on

$$N_i = I_{n_i} + X_i^T X_i \in \mathbb{R}^{n_i\times n_i}. \tag{25}$$

However, the inversion M_i^{-1} will not be formed explicitly, as the multiplication of $(\tilde{x}-\tilde{\mu}_i)^T M_i^{-1}(\tilde{x}-\tilde{\mu}_i)$, takes $O(n^2 k)$ time, for each x. Note from Eq. (24) that

$$\begin{aligned} &(\tilde{x}-\tilde{\mu}_i)^T M_i^{-1}(\tilde{x}-\tilde{\mu}_i) = (\tilde{x}-\tilde{\mu}_i)^T D_{\alpha\beta}^{-1}(\tilde{x}-\tilde{\mu}_i) - \\ &(\tilde{x}-\tilde{\mu}_i)^T D_{\alpha\beta}^{-0.5} X_i N_i^{-1} X_i^T D_{\alpha\beta}^{-0.5}(\tilde{x}-\tilde{\mu}_i), \end{aligned} \tag{26}$$

which takes $O(n_i^3)$ for computing N_i^{-1} and $O(nn_i)$ time for all other computations, for each x. The total complexity is thus $O(nn_i + n_i^3)$ time, for each x. Thus, the computation of $(\tilde{x}-\tilde{\mu}_i)^T M_i^{-1}(\tilde{x}-\tilde{\mu}_i)$ for all i takes $O(n^2 + \sum_{i=1}^k n_i^3)$ time. Assuming all classes are of approximately equal size, that is, $n_i \approx n/k$, then the time complexity for the computation is $O(n^2 + n^3/k^2)$.

One key observation here is that the computation of N_i^{-1} is independent of the test point x. Note that the total number of test points in v-fold cross-validation is n/v. In this case, the total computational cost for all test points is $O(n^3 + n^3/k^2)$, instead of $O(n^3 + n^4/k^2)$.

Next, we consider the computation of $|M_i|$, which is independent of the test point x. From Eq. (23),

$$|M_i| = |X_i X_i^T + I_t||D_{\alpha\beta}| = |X_i^T X_i + I_{n_i}||D_{\alpha\beta}|, \tag{27}$$

where the last equality follows from the following lemma:

LEMMA 3.4. *Let $X \in \mathbb{R}^{t\times n_i}$ be any matrix of size t by n_i. Then the following equality always holds:*

$$| XX^T + I_t | = | X^T X + I_{n_i} | .$$

PROOF. Let $X = RDS^T$ be the SVD of X, where R and S are orthogonal and $D = \begin{pmatrix} \Sigma_m & 0 \\ 0 & 0 \end{pmatrix}$ is diagonal with $\Sigma_m = \mathrm{diag}(\lambda_1, \cdots, \lambda_m)$ and $m = \mathrm{rank}(X)$. It follows that

$$\begin{aligned} |XX^T + I_t| &= |R(DD^T + I_t)R^T| = |DD^T + I_t| \\ &= |\Sigma_m^2 + I_m| \cdot |I_{t-m}| = \prod_{j=1}^m (\lambda_j^2 + 1), \\ |X^T X + I_{n_i}| &= |S(D^T D + I_{n_i})S^T| = |D^T D + I_{n_i}| \\ &= |\Sigma_m^2 + I_m| \cdot |I_{n_i - m}| = \prod_{j=1}^m (\lambda_j^2 + 1), \end{aligned}$$

Thus $| XX^T + I_t | = | X^T X + I_{n_i} |$. □

From Eq. (27), the time complexity of the computation of $|M_i|$, for all i, is

$$O\left(t\sum_{i=1}^k n_i^2 + \sum_{i=1}^k n_i^3\right) = O\left(n\sum_{i=1}^k n_i^2 + \sum_{i=1}^k n_i^3\right),$$

which is $O(n^3/k)$, assuming all classes are of approximately equal size.

3.2 The computation of U_1

Recall that the key matrix in the decomposition property of RDA in Proposition 3.1 is U_1, which lies in the orthogonal complement of the null space of S_t. We have applied the SVD for computing U_1 as $S_t = UDU^T$, where $U = [U_1, U_2]$ is a partition of U. When the data dimensionality d is large, the full SVD computation of $S_t \in \mathbb{R}^{d\times d}$ is expensive. However from Lemma 3.3, only the first component of the classification rule, which involves U_1, is effective in RDA, while U_2, the null space of S_t can simply be omitted. Thus, U_1 can be computed efficiently without the full SVD computation of S_t as follows. Define matrix H_t as:

$$H_t = \frac{1}{\sqrt{n}}(A - \mu e^T), \tag{28}$$

where μ is the global centroid and e is the vector of all ones. It follows from the definition that $S_t = H_t H_t^T$. Note that $H_t \in \mathbb{R}^{d\times n}$, which is much smaller than S_t for HDLSS data. Let $H_t = \hat{U}\hat{\Sigma}\hat{V}^T$ be the reduced SVD of H_t, where $\hat{U} \in \mathbb{R}^{d\times t}$ and $\hat{V} \in \mathbb{R}^{n\times t}$ have orthonormal columns and $\hat{\Sigma} \in \mathbb{R}^{t\times t}$ is diagonal with $t = \mathrm{rank}(H_t)$. It follows that $S_t = H_t H_t^T = \hat{U}\hat{\Sigma}^2\hat{U}^T$. Thus, $U_1 = \hat{U}$ and $D_t = \hat{\Sigma}^2$. The time complexity of the reduced SVD computation of H_t is $O(dn^2)$ [10], instead of $O(d^3)$ for the full SVD computation.

3.3 The main algorithm

Let $\Lambda = \{(\alpha_i, \beta_j)\}$, where $i = 1, \cdots, r$ and $j = 1, \cdots, s$, be the candidate set for the regularization parameters. In model selection, v-fold cross-validation is applied, where the data is divided into v subsets of (approximately) equal size. All subsets are mutually exclusive, and in the i-th fold, the i-th subset is held out for test and all other subsets are used in training. For each (α_i, β_j), we compute the cross-validation accuracy, Accu(i, j), defined as the mean of the accuracies for all folds. The best regularization pair (α_i^*, β_j^*) is the one with $(i^*, j^*) = \arg\max_{i,j} \text{Accu}(i, j)$. The pseudo-code of the proposed algorithm is given below.

Algorithm RDA
Input: data matrix A
set of parameters: $\{\alpha_i\}_{i=1}^r$ and $\{\beta_j\}_{j=1}^s$
Output: the optimal parameter pair $(\alpha_{i^*}, \beta_{j^*})$

1. For $h = 1 : v$ /* v-fold cross validation */
2. Construct A^h and $A^{\hat{h}}$;
   /* A^h = h-th fold, for training */
   /* $A^{\hat{h}}$ = rest, for testing */
3. Construct H_t using A^h as in Eq. (28);
4. Compute the reduced SVD of H_t: $H_t = \hat{U}\hat{\Sigma}\hat{V}^T$;
5. $t \leftarrow \text{rank}(H_t)$; $U_1 \leftarrow \hat{U}$; $D_t \leftarrow \hat{\Sigma}^2$;
6. $A_L^h \leftarrow U_1^T A^{\hat{h}}$; $A_L^{\hat{h}} \leftarrow U_1^T A^{\hat{h}}$;
   /* Null space, U_2, of S_t is removed */
7. Form $\{\tilde{H}_u\}_{u=1}^k$ based on A_L^h as in Eq. (19);
8. For $i = 1 : r$ /* $\alpha_1, \alpha_2, \cdots, \alpha_r$ */
9. For $j = 1 : s$ /* $\beta_1, \beta_2, \cdots, \beta_s$ */
10. $D_{\alpha\beta} \leftarrow (1 - \alpha_i)\beta_j D_t + (1 - \beta_j) I_t$;
11. For $u = 1 : k$
12. $X_u \leftarrow \sqrt{\alpha_i \beta_j} D_{\alpha\beta}^{-0.5} \tilde{H}_u$;
13. $N_i^{-1} \leftarrow (I + X_u^T X_u)^{-1}$ as in Eq. (25),
14. Compute $|M_u|$ as in Eq. (27),
15. EndFor
16. temp $\leftarrow 0$;
    /* Variable temp counts the number of test points correctly classified */
17. For each $\tilde{x} \in A_L^{\hat{h}}$ /* $\tilde{x} = U_1^T x$ and $x \in A^{\hat{h}}$ */
18. $C(x) \leftarrow \text{argmin}_u \{(x - \tilde{\mu}_u)^T M_u^{-1} (x - \tilde{\mu}_u)$
19. $+\ln|M_u|\}$;
    /* The multiplication is done as in Eq. (26) */
20. If $(C(x) == g(x))$ temp $\leftarrow$ temp $+1$;
21. EndFor
22. Accu$(h, i, j) \leftarrow$ temp$/|A_L^{\hat{h}}|$;
    /* $|A_L^{\hat{h}}|$ denotes the number of test points */
23. EndFor
24. EndFor
25. EndFor
26. Accu$(i, j) \leftarrow \frac{1}{v} \sum_{h=1}^v$ Accu(h, i, j); /* Accu(i, j) denotes the cross-validation accuracy */
27. $(i^*, j^*) \leftarrow \arg\max_{i,j}$ Accu(i, j);
28. Output $(\alpha_{i^*}, \beta_{j^*})$ as the best parameter pair.

3.4 Time Complexity

Line 4 takes $O(n^2 d)$ time for the reduced SVD computation [10]. Lines 5 and 6 take $O(dn^2)$ time for the matrix multiplications. The "For" loop from Line 11 to Line 15 takes $O(n^3/k)$ time. There are about n/v elements in $A_L^{\hat{h}}$, thus Line 18 to Line 20 within the "For" loop run about n/v times. Following the multiplication in Eq. (26), the computations from Line 18 to Line 20 take $O(n^2)$ time, and the "For" loop from Line 17 to Line 21 take $O(n^3/v)$ time. Thus, the double "For" loops from Line 8 to Line 26 take $O\left(n^3 rs(1/k + 1/v)\right)$ time. The total running time of the algorithm is thus

$$\begin{aligned} T(r, s) &= O\left(v(n^2 d + n^3 rs(1/k + 1/v))\right) \\ &= O\left(vn^2(d + nrs(1/k + 1/v))\right). \end{aligned}$$

It follows that

$$\frac{T(r, s)}{T(1, 1)} \approx \frac{d + nrs(1/k + 1/v)}{d + n(1/k + 1/v)} \approx 1 + \frac{nrs(1/k + 1/v)}{d + n(1/k + 1/v)}.$$

For HDLSS data, where the sample size n is much smaller than the data dimensionality d, i.e., $n \ll d$, the overhead of estimating the optimal regularization pair among a large search space may be small.

Note that the first stage of RDA takes $O(vn^2 d)$ time, which is expensive for HDLSS data. However, it is independent of the parameters. In the second stage of RDA, the most expensive steps are the computations of M_i^{-1} and $|M_i|$, which take $O\left(vn^3 rs(1/k + 1/v)\right)$ time. It is independent of the data dimensionality d. This is the key reason why the proposed RDA algorithm is applicable for HDLSS data for a large candidate set of parameters.

3.5 RDA versus ULDA

We conclude this section by showing an interesting relationship between RDA and Uncorrelated LDA (LDA) [23]. ULDA is an extension of the original formulation in [16] to high dimensional, small sample size data. It follows the basic framework of LDA [5, 7, 9, 14] that computes the optimal transformation (projection) by minimizing the ratio of the within-class distance to the between-class distance, thus achieving maximum discrimination. One key property of ULDA is that the features in the transformed space are uncorrelated, thus ensuring minimum redundancy among the features in the reduced space. It has been applied successfully in microarray gene expression data analysis [24]. It was shown in [22] that the optimal transformation G of ULDA consists of the first q eigenvectors of $S_t^+ S_b$, where $q = \text{rank}(S_b)$. With the computed G, a test point x can be classified in ULDA as class h, where $h = \arg\min_i \left\{||G^T(x - \mu_i)||^2\right\}$. It has been shown in [22] that

$$\arg\min_i \left\{(x - \mu_i)^T S_t^+ (x - \mu_i)\right\} = \arg\min_i \left\{||G^T(x - \mu_i)||^2\right\}.$$

Interestingly, we can show that the limit of RDA when $\alpha \to 0$ and $\beta \to 1$ is equivalent to ULDA as summarized in the following lemma:

THEOREM 3.1. *The classification rule in RDA approaches that of ULDA, when $\alpha \to 0$ and $\beta \to 1$. That is, if G is the transformation of ULDA. Then,*

$$\lim_{\alpha \to 0, \beta \to 1} \hat{C}(x) = \arg\min_i \left\{||G^T(x - \mu_i)||^2\right\}.$$

PROOF. From Eq. (18), the classification rule in RDA is equivalent to

$$\hat{C}(x) = \text{argmin}_i \left\{(\tilde{x} - \tilde{\mu}_i)^T M_i^{-1} (\tilde{x} - \tilde{\mu}_i) + \ln|M_i|\right\}, \quad (29)$$

dataset	size (n)	dimensionality (d)	# of classes (k)
re0	320	2887	4
re1	490	3759	5
ORL	400	10304	40
PIX	300	10000	30
ALL	248	12558	6
ALLAML4	72	7129	4

Table 2: Statistics for our test datasets.

From Eq. (14), we have $\lim_{\alpha\to 0,\beta\to 1} M_i = D_t$. Thus

$$\begin{aligned}\lim_{\alpha\to 0,\beta\to 1}\hat{C}(x) &= \text{argmin}_i\left\{(\tilde{x}-\tilde{\mu}_i)^T D_t^{-1}(\tilde{x}-\tilde{\mu}_i)+\ln|D_t|\right\},\\ &= \text{argmin}_i\left\{(x-\mu_i)^T U_1 D_t^{-1} U_1^T (x-\mu_i)\right\}\\ &= \text{argmin}_i\left\{(x-\mu_i)^T S_t^+ (x-\mu_i)\right\},\end{aligned}$$

which is equivalent to

$$\arg\min_i\left\{||G^T(x-\mu_i)||^2\right\},$$

the classification rule in ULDA. □

Theorem 3.1 shows that ULDA is a special case of RDA when $\alpha = 0$ and $\beta = 1$. With a properly chosen parameter pair (α, β) through cross-validation, RDA is expected to outperform ULDA, which is confirmed by the empirical results presented in the next section.

Note that the limit of $\hat{\Sigma}_i^{-1}$, as $\alpha \to 0$ and $\beta \to 1$ does not exist for HDLSS data, as the limit of

$$\hat{\Sigma}_i = \beta\left(\alpha\Sigma_i + (1-\alpha)S_t\right) + (1-\beta)I_d$$

is singular, when $d > n$. However, Theorem 3.1 shows that the limit below exists:

$$\lim_{\alpha\to 0,\beta\to 1}(x-\mu_i)^T\hat{\Sigma}_i^{-1}(x-\mu_i) = (x-\mu_i)^T S_t^+(x-\mu_i),$$

due to the decomposition property of RDA as in Eq. (18).

4. EXPERIMENTS

In this section, we experimentally evaluate the performance of the proposed RDA algorithm. v-fold cross validation with $v = 5$ has been used in RDA for model selection. All of our experiments have been performed on a P4 3.00GHz Windows XP machine with 2GB memory.

4.1 Datasets

We have used three types of HDLSS data for the evaluation, including text documents, face images, and gene expression data. The important statistics of these datasets are summarized below (see also Table 2):

- re0 and re1 are two text document datasets, derived from *Reuters-21578* text categorization test collection Distribution 1.0 [18]. re0 includes 320 documents belonging to 4 different classes. The dimension of this dataset is 2887. re1 has 5 classes, each with 98 instances; its dimension is 3759.
- ORL is a face image dataset, which contains 400 face images of 40 individuals. The image size is 92×112. The face images are perfectly centralized. The major challenge on this dataset is the variation of the face pose. There is no lighting variation, with minimal facial expression variations, and no occlusion. We use the whole image as an instance (i.e., the dimension of an instance is $92 \times 112 = 10304$).
- PIX is a face image dataset, which contains 300 face images of 30 individuals. The image size is 512×512. We subsample the images with a sample step of 5×5, and the dimension of each instance is reduced to $100 \times 100 = 10000$.
- ALL is a gene expression dataset consisting of six diagnostic groups [25]. The breakdown of the samples is: 15 samples for *BCR*, 27 samples for *E2A*, 64 samples for *Hyperdip*, 20 samples for *MLL*, 43 samples for *T*, and 79 samples for *TEL*.
- ALLAML4 is a gene expression dataset, which contains the gene expression profiles of two acute cases of leukemia: acute lymphoblastic leukemia (ALL) and acute myeloblastic leukemia (AML). The ALL part of the dataset comes from two sample types, B-cell and T-cell, and the AML part is split into bone marrow samples and peripheral blood. This dataset was fist studied in the seminal paper of Golub *et al.* [11]. Golub *et al.* studied this problem to address the binary classification problem between the AML samples and the ALL samples. ALLAML4 is a four-class dataset (*B-cell*, *T-cell*, *AML-BM*, and *AML PB*).

4.2 Efficiency

In this experiment, we test the efficiency of the proposed RDA algorithm. Table 3 shows the computational time (in seconds) of RDA on different numbers of parameter pairs $r \times s$. We set $r = s$ for simplicity with r taking values from 1 to 32, thus the size of the candidate set, $|\Lambda|$ ranges from 1 to 1024. It is clear from the table that the computational cost of RDA grows slowly as $r \times s$ is small. When $r \times s$ is large, the cost, $T(r, s)$, of the proposed RDA algorithm is still significantly smaller than $rsT(1, 1)$, the computational cost of RDA without applying the optimizations proposed in this paper.[1] For example, we can observe that $T(16, 16)/T(1, 1)$ on different datasets is less than 7, which is significantly smaller than $16 \times 16 = 256$, while $T(32, 32)/T(1, 1)$ is less than 25 in all cases, much smaller than $32 \times 32 = 1024$. Among all datasets, the document datasets have relatively larger increasing rates of running time than the others, while the gene expression datasets have the smallest increasing rates. Note that the ratio of the sample size to the data dimensionality, i.e., n/d, is relatively large for both document datasets, while it is relatively small for both gene expression datasets. These results are consistent with the theoretical estimation of the efficiency in Section 3.

4.3 Classification performance

In this experiment, we evaluate RDA in classification and compare it with Uncorrelated LDA [23] and Support Vector Machines (SVM) [2, 4, 21]. For each dataset, we first set the percentage of the data for training to be either 1/2 or 1/3 (by a random partition). Then we apply the proposed RDA algorithm, as well as ULDA and SVM, on the training data to learn the model, which is further applied to the remaining

[1]Note that the cost of RDA will be even higher if the decomposition property of RDA from this paper is not applied.

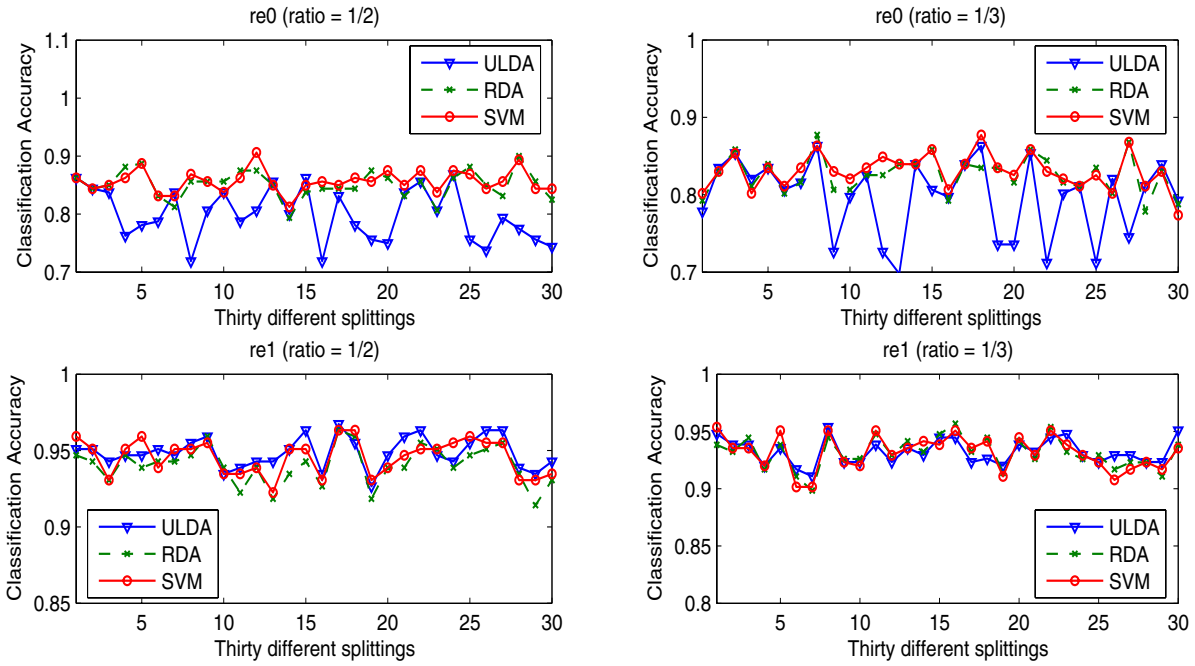

Figure 1: Comparison of ULDA, RDA, and SVM in classification accuracy using re0 and re1. The x-axis denotes 30 different partitions into training and testing set, where ratio is the percentage of the data for training.

	# of parameter pairs ($r \times s$)					
dataset	1×1	2×2	4×4	8×8	16×16	32×32
re0	0.97	1.05	1.17	1.66	4.52	19.2
re1	2.19	2.30	2.75	4.58	13.5	54.6
ORL	3.59	3.91	5.13	10.2	24.7	73.2
PIX	2.17	2.39	2.95	5.42	14.5	46.0
ALL	2.36	2.39	2.47	3.14	11.2	36.5
ALLAML4	0.17	0.19	0.21	0.25	0.63	1.84

Table 3: Computational time (in seconds) of RDA for different numbers of parameter pairs.

test data to get the accuracy of classification. To give a better estimation of accuracy, the procedure is repeated 30 times and the resulting accuracies are averaged. Note that for RDA, we choose the optimal model from 900 parameter pairs with $r = 30$ and $s = 30$. Because of the improved efficiency of the proposed RDA algorithm, it is practical to select the optimal model from such a large search space.

The classification accuracies of the 30 different partitions for all six datasets are shown in Fig. 1–3. In Table 4, we report the mean accuracy and standard derivation of the 30 different partitions for each dataset, where *ratio* denotes the percentage of the data for training and is either $1/3$ or $1/2$ in our experiments. For all the datasets, the performance using $ratio = 1/2$ is better than that using $ratio = 1/3$ in terms of classification accuracy. This conforms to our expectation that the classification performance may be improved with a larger number of training data. We can observe from the accuracy curves in Fig. 1–3 that RDA and SVM often follow similar trends.

For both document datasets re1 and re0, all three algorithms achieve comparable performance in re1 (all three accuracy curves in Fig. 1 are very close to each other), while RDA and SVM outperform ULDA in re0. For both face image datasets, RDA and SVM outperform ULDA by a large margin, while RDA achieves slightly higher accuracies than SVM. As for both gene expression datasets, the accuracy curves are similar for all three algorithms, while RDA achieves a smaller overall variance than ULDA and SVM. Overall, RDA is very competitive with ULDA and SVM in classification. Recall from Theorem 3.1 that ULDA is a special case of RDA when $\alpha = 0$ and $\beta = 1$. With a properly chosen parameters, RDA is expected to outperform ULDA, which is confirmed by our empirical results above.

5. CONCLUSIONS

We present in this paper a novel algorithm for RDA that is applicable for high dimensional, low sample size data. RDA is a compromise between LDA and QDA, regulated by two regularization parameters. A major advantage of the proposed RDA algorithm is its low computational cost in selecting the optimal parameters from a large candidate set, in comparison with the traditional RDA formulation. Thus it facilitates efficient model selection for RDA. The key to the proposed efficient model selection procedure lies in the decomposition property of RDA established in this paper. We evaluate the proposed algorithm using document, image, and gene expression datasets. RDA is compared with ULDA and SVM in classification. Results confirm the high efficiency of the proposed algorithm. Our experiments also demonstrate that with the proposed efficient model selection algorithm, RDA can be effectively applied to high dimensional, low sample size data.

The relative performance of RDA over ULDA varies a lot for different types of data. RDA outperforms ULDA for both

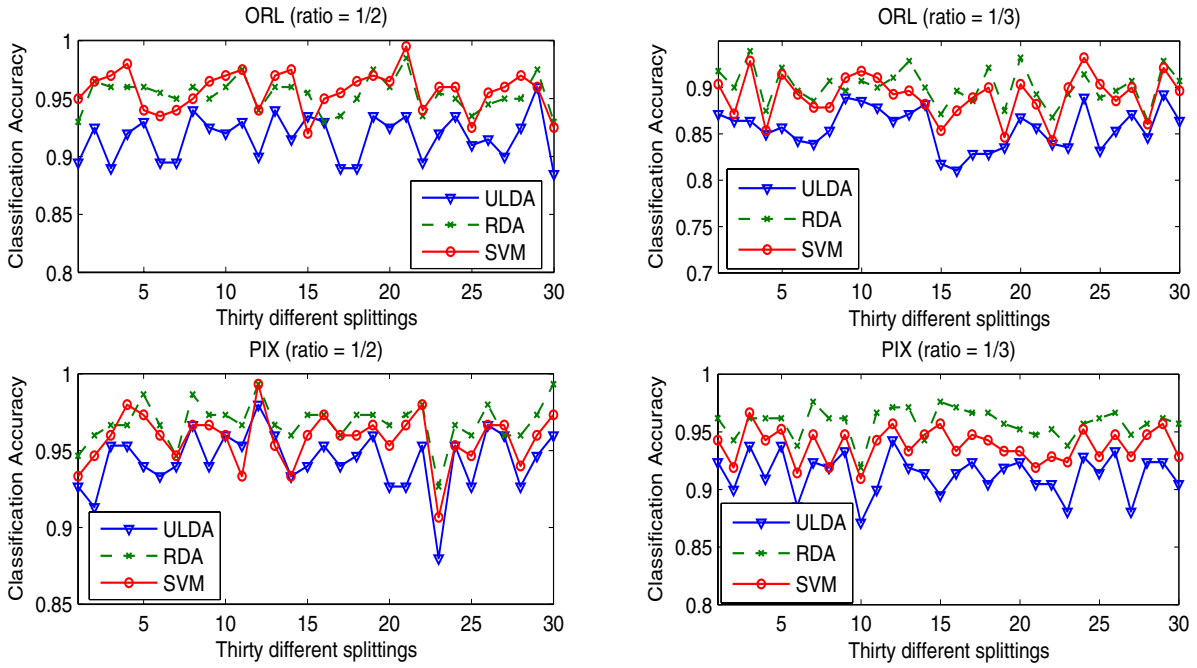

Figure 2: Comparison of ULDA, RDA, and SVM in classification accuracy using ORL and PIX. The x-axis denotes 30 different partitions into training and testing set, where ratio is the percentage of the data for training.

image datasets by a large margin, while they are comparable for both gene expression datasets. One of the future work is to study the effect of the characteristics of the data on the performance of RDA. We also plan to apply RDA to other applications involving HDLSS data such as gene expression pattern images, protein expression data, etc.

Table 4: Comparison of classification accuracy (in percentage) of ULDA, RDA, and SVM. The mean and standard deviation of 30 different partitions with a ratio of 1/2 and 1/3 are reported.

		ULDA		RDA		SVM	
dataset	ratio	mean	std	mean	std	mean	std
re0	1/2	79.85	4.51	85.12	2.41	85.67	1.96
re0	1/3	79.67	4.95	82.59	2.46	83.07	2.28
re1	1/2	94.88	1.02	94.05	1.25	94.59	1.16
re1	1/3	93.23	1.11	93.14	1.38	93.14	1.51
ORL	1/2	91.68	1.93	95.33	1.42	95.67	1.76
ORL	1/3	85.62	2.20	90.10	1.95	89.08	2.35
PIX	1/2	94.40	1.95	96.84	1.38	95.80	1.73
PIX	1/3	91.33	1.81	95.79	1.27	93.84	1.45
ALL	1/2	97.24	1.28	97.87	1.02	97.19	1.00
ALL	1/3	95.38	1.44	96.31	1.21	95.58	1.76
ALLAML4	1/2	93.03	2.39	93.24	2.05	92.57	2.32
ALLAML4	1/3	88.42	2.82	88.94	3.08	88.69	2.70

Acknowledgements

Research of JY is sponsored, in part, by the Center for Evolutionary Functional Genomics of the Biodesign Institute at the Arizona State University.

6. REFERENCES

[1] P.N. Belhumeour, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

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

[3] L.F. Chen, H.Y.M. 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*, 33:1713–1726, 2000.

[4] N. Cristianini and J.S. Taylor. *Support Vector Machines and other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[5] R.O. Duda, P.E. Hart, and D. Stork. *Pattern Classification*. Wiley, 2000.

[6] S. Dudoit, J. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457):77–87, 2002.

[7] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

[8] J.H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

[9] K. Fukunaga. *Introduction to Statistical Pattern Classification*. Academic Press, USA, 1990.

[10] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, USA, third edition, 1996.

[11] T.R. Golub and et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

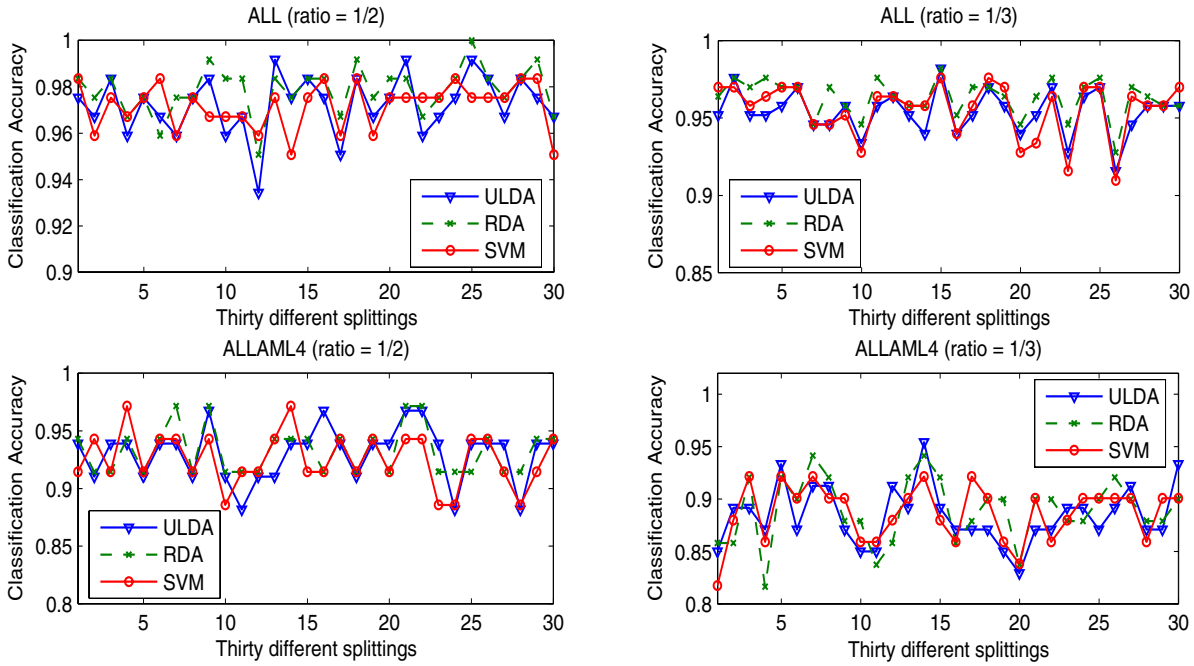

Figure 3: Comparison of ULDA, RDA, and SVM in classification accuracy using ALL and ALLAML4. The x-axis denotes 30 different partitions into training and testing set, where ratio is the percentage of the data for training.

[12] U. Grouven, F. Bergel, and A. Schultz. Implementation of linear and quadratic discriminant analysis incorporating costs of misclassification. *Computer Methods and Programs in Biomedicine*, 49(1):55–60, 1996.

[13] P. Hall, J.S. Marron, and A. Neeman. Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society series B*, 67:427–444, 2005.

[14] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer, 2001.

[15] A. Hoerl and R. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(3):55–67, 1970.

[16] Z. Jin, J. Y. Yang, Z.S. Hu, and Z. Lou. Face recognition based on the uncorrelated discriminant transformation. *Pattern Recognition*, 34:1405–1416, 2001.

[17] W.J. Krzanowski, P. Jonathan, W.V McCarthy, and M.R. Thomas. Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data. *Applied Statistics*, 44:101–115, 1995.

[18] D.D. Lewis. *Reuters-21578 text categorization test collection distribution 1.0*. http://www.research.att.com/∼lewis, 1999.

[19] D. L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans Pattern Analysis and Machine Intelligence*, 18(8):831–836, 1996.

[20] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-posed problems*. John Wiley and Sons, Washington D.C., 1977.

[21] V.N. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[22] J. Ye. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *Journal of Machine Learning Research*, 6:483–502, 2005.

[23] J. Ye, R. Janardan, Q. Li, and H. Park. Feature extraction via generalized uncorrelated linear discriminant analysis. In *ICML Conference Proceedings*, 2004.

[24] J. Ye, T. Li, T. Xiong, and R. Janardan. Using uncorrelated discriminant analysis for tissue classification with gene expression data. *IEEE/ACM Trans. Computational Biology and Bioinformatics*, 1(4):181–190, 2004.

[25] E.J. Yeoh et al. Classification, subtype discovery, and prediction of outcome in pediatric lymphoblastic leukemia by gene expression profiling. *Cancer Cell*, 1(2):133–143, 2002.

[26] L. Zhang and L. Luo. Splice site prediction with quadratic discriminant analysis using diversity measure. *Nucleic Acids Research*, 31(21):6214–6220, 2003.

[27] M. Zhang. Identification of protein coding regions in the human genome by quadratic discriminant analysis. *Proceedings of the National Academy of Sciences, USA*, 94:565–568, 1997.