

Chapter 1

Discriminant Analysis for Dimensionality Reduction: An Overview of Recent Developments

Jieping Ye and Shuiwang Ji

1.1 INTRODUCTION

Many biometric applications such as face recognition involve data with a large number of features [1–3]. Analysis of such data is challenging due to the *curse-of-dimensionality* [4, 5], which states that an enormous number of samples are required to perform accurate predictions on problems with a high dimensionality. Dimensionality reduction, which extracts a small number of features by removing irrelevant, redundant, and noisy information, can be an effective solution [6]. The commonly used dimensionality reduction methods include supervised approaches such as linear discriminant analysis (LDA) [7, 8], unsupervised ones such as principal component analysis (PCA) [9], and additional spectral and manifold learning methods [10–14]. When the class label information is available, supervised approaches, such as LDA, are usually more effective than unsupervised ones such as PCA for classification.

Linear discriminant analysis (LDA) is a classical statistical approach for supervised dimensionality reduction and classification [8, 15–18]. LDA computes an optimal transformation (projection) by minimizing the within-class distance and maximizing the between-class distance simultaneously, thus achieving maximum

class discrimination. The optimal transformation in LDA can be readily computed by applying an eigendecomposition on the so-called scatter matrices. It has been used widely in many applications involving high-dimensional data [19–24]. However, classical LDA requires the so-called *total scatter matrix* to be nonsingular. In many applications involving high-dimensional and low sample size data, the total scatter matrix can be singular since the data points are from a very high-dimensional space, and in general the sample size does not exceed this dimension. This is the well-known *singularity or undersampled problem* encountered in LDA.

In recent years, many LDA extensions have been proposed to deal with the singularity problem, including PCA+LDA [19, 23], regularized LDA (RLDA) [21], null space LDA (NLDA) [20], orthogonal centroid method (OCM) [25], uncorrelated LDA (ULDA) [24], orthogonal LDA (OLDA) [24], and LDA/GSVD [26]. A brief overview of these algorithms is given in Section 1.2. Different algorithms have been applied successfully in various domains, such as PCA+LDA in face recognition [19, 23], OCM in text categorization [25], and RLDA in microarray gene expression data analysis [21]. However, there is a lack of a systematic study to explore the commonalities and differences of these algorithms, as well as their intrinsic relationship. This has been a challenging task, since different algorithms apply completely different schemes when dealing with the singularity problem.

Many of these LDA extensions involve an eigenvalue problem, which is computationally expensive to solve especially when the sample size is large. LDA in the binary-class case, called Fisher LDA, has been shown to be equivalent to linear regression with the class label as output. Such regression model minimizes the sum-of-squares error function whose solution can be obtained efficiently by solving a system of linear equations. However, the equivalence relationship is limited to the binary-class case.

In this chapter, we present a unified framework for generalized LDA via a transfer function. We show that various LDA-based algorithms differ in their transfer functions. The unified framework elucidates the properties of various algorithms and their relationship. We then discuss recent development on establishing the equivalence relationship between multivariate linear regression (MLR) and LDA in the multiclass case. In particular, we show that MLR with a particular class indicator matrix is equivalent to LDA under a mild condition, which has been shown to hold for most high-dimensional data. We further show how LDA can be performed in the semisupervised setting, where both labeled and unlabeled data are provided, based on the equivalence relationship between MLR and LDA. We also extend our discussion to the kernel-induced feature space and present recent developments on multiple kernel learning (MKL) for kernel discriminant analysis (KDA).

The rest of this chapter is organized as follows. We give an overview of classical LDA and its generalization in Section 1.2. A unified framework for generalized LDA as well as the theoretical properties of various algorithms and their relationship is presented in Section 1.3. Section 1.4 discusses the least squares formulation for LDA. We then present extensions of the discussion to semisupervised learning and kernel-induced feature space in Sections 1.5 and 1.6, respectively. This chapter concludes in Section 1.8.

1.2 OVERVIEW OF LINEAR DISCRIMINANT ANALYSIS

We are given a data set that consists of n samples $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ denotes the d -dimensional input, $y_i \in \{1, 2, \dots, k\}$ denotes the corresponding class label, n is the sample size, and k is the number of classes. Let

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

be the data matrix and let $X_j \in \mathbb{R}^{d \times n_j}$ be the data matrix of the j th class, where n_j is the sample size of the j th class and $\sum_{j=1}^k n_j = n$. Classical LDA computes a linear transformation $G \in \mathbb{R}^{d \times \ell}$ that maps x_i in the d -dimensional space to a vector x_i^L in the ℓ -dimensional space as follows:

$$x_i \in \mathbb{R}^d \rightarrow x_i^L = G^T x_i \in \mathbb{R}^\ell, \quad \ell < d.$$

In LDA, three scatter matrices, called the *within-class*, *between-class*, and *total* scatter matrices are defined as follows [8]:

$$S_w = \frac{1}{n} \sum_{j=1}^k \sum_{x \in X_j} (x - c^{(j)})(x - c^{(j)})^T, \quad (1.1)$$

$$S_b = \frac{1}{n} \sum_{j=1}^k n_j (c^{(j)} - c)(c^{(j)} - c)^T, \quad (1.2)$$

$$S_t = \frac{1}{n} \sum_{i=1}^n (x_i - c)(x_i - c)^T, \quad (1.3)$$

where $c^{(j)}$ is the *centroid* of the j th class and c is the *global centroid*. It can be verified from the definitions that $S_t = S_b + S_w$ [8]. Define three matrices H_w , H_b , and H_t as follows:

$$H_w = \frac{1}{\sqrt{n}} [X_1 - c^{(1)}(e^{(1)})^T, \dots, X_k - c^{(k)}(e^{(k)})^T], \quad (1.4)$$

$$H_b = \frac{1}{\sqrt{n}} [\sqrt{n_1}(c^{(1)} - c), \dots, \sqrt{n_k}(c^{(k)} - c)], \quad (1.5)$$

$$H_t = \frac{1}{\sqrt{n}} (X - ce^T), \quad (1.6)$$

where $e^{(j)}$ and e are vectors of all ones of length n_j and n , respectively. Then the three scatter matrices, defined in Eqs. (1.1)–(1.3), can be expressed as

$$S_w = H_w H_w^T, \quad S_b = H_b H_b^T, \quad S_t = H_t H_t^T. \quad (1.7)$$

It follows from the properties of matrix trace that

$$\text{trace}(S_w) = \frac{1}{n} \sum_{j=1}^k \sum_{x \in X_j} \|x - c^{(j)}\|_2^2, \quad (1.8)$$

$$\text{trace}(S_b) = \frac{1}{n} \sum_{j=1}^k n_j \|c^{(j)} - c\|_2^2. \quad (1.9)$$

Thus $\text{trace}(S_w)$ measures the distance between the data points and their corresponding class centroid, and $\text{trace}(S_b)$ captures the distance between the class centroids and the global centroid.

In the lower-dimensional space resulting from the linear transformation G , the scatter matrices become

$$S_w^L = G^T S_w G, \quad S_b^L = G^T S_b G, \quad S_t^L = G^T S_t G. \quad (1.10)$$

An optimal transformation G would maximize $\text{trace}(S_b^L)$ and minimize $\text{trace}(S_w^L)$ simultaneously, which is equivalent to maximizing $\text{trace}(S_b^L)$ and minimizing $\text{trace}(S_t^L)$ simultaneously, since $S_t^L = S_w^L + S_b^L$. The optimal transformation, G^{LDA} , of LDA is computed by solving the following optimization problem [8, 16]:

$$G^{LDA} = \arg \max_G \left\{ \text{trace} \left(S_b^L (S_t^L)^{-1} \right) \right\}. \quad (1.11)$$

It is known that the optimal solution to the optimization problem in Eq. (1.11) can be obtained by solving the following generalized eigenvalue problem [8]:

$$S_b x = \lambda S_t x. \quad (1.12)$$

More specifically, the eigenvectors corresponding to the $k - 1$ largest eigenvalues form columns of G^{LDA} . When S_t is nonsingular, it reduces to the following regular eigenvalue problem:

$$S_t^{-1} S_b x = \lambda x. \quad (1.13)$$

When S_t is singular, the classical LDA formulation discussed above cannot be applied directly. This is known as the *singularity* or *undersampled* problem in LDA. In the following discussion, we consider the more general case when S_t may be singular. The transformation, G^{LDA} , then consists of the eigenvectors of $S_t^+ S_b$ corresponding to the nonzero eigenvalues, where S_t^+ denotes the pseudo-inverse of S_t [27]. Note that when S_t is nonsingular, S_t^+ equals S_t^{-1} .

The above LDA formulation is an extension of the original Fisher linear discriminant analysis (FLDA) [7], which deals with binary-class problems, that is, $k = 2$. The optimal transformation, G^F , of FLDA is of rank one and is given by [15, 16]

$$G^F = S_t^+ (c^{(1)} - c^{(2)}). \quad (1.14)$$

Note that G^F is invariant of scaling. That is, αG^F , for any $\alpha \neq 0$, is also a solution to FLDA.

When the dimensionality of data is larger than the sample size, which is the case for many high-dimensional and low sample size data, all of the three scatter matrices are singular. In recent years, many algorithms have been proposed to deal with this singularity problem. We first review these LDA extensions in the next subsection. To

elucidate their commonalities and differences, a general framework is presented in Section 1.3 that unifies many of these algorithms.

1.2.1 Generalizations of LDA

A common way to deal with the singularity problem is to apply an intermediate dimensionality reduction, such as PCA [9], to reduce the data dimensionality before classical LDA is applied. The algorithm is known as PCA+LDA, or subspace LDA [19, 28]. In this two-stage PCA+LDA algorithm, the discriminant stage is preceded by a dimensionality reduction stage using PCA. The dimensionality, p , of the subspace transformed by PCA is chosen such that the “reduced” total scatter matrix in this subspace is nonsingular, so that classical LDA can be applied. The optimal value of p is commonly estimated through cross-validation.

Regularization techniques can also be applied to deal with the singularity problem of LDA. The algorithm is known as regularized LDA (RLDA) [21]. The key idea is to add a constant $\mu > 0$ to the diagonal elements of S_t as $S_t + \mu I_d$, where I_d is the identity matrix of size d . It is easy to verify that $S_t + \mu I_d$ is positive definite [27], hence nonsingular. Cross-validation is commonly applied to estimate the optimal value of μ . Note that regularization is also the key to many other learning algorithms including Support Vector Machines (SVM) [29].

In reference 20, the null space LDA (NLDA) was proposed, where the between-class distance is maximized in the null space of the within-class scatter matrix. The singularity problem is thus avoided implicitly. The efficiency of the algorithm can be improved by first removing the null space of the total scatter matrix. It is based on the observation that the null space of the total scatter matrix is the intersection of the null spaces of the between-class and within-class scatter matrices. In contrast, the orthogonal centroid method (OCM) [25] maximizes the between-class distance only and thereby omits the within-class information. The optimal transformation of OCM is given by the top eigenvectors of the between-class scatter matrix S_b .

In reference 24, a family of generalized discriminant analysis algorithms were presented. Uncorrelated LDA (ULDA) and orthogonal LDA (OLDA) are two representative algorithms from this family. The features in the reduced space of ULDA are uncorrelated, while the transformation, G , of OLDALDA has orthonormal columns, that is, $G^T G = I_\ell$. The LDA/GSVD algorithm proposed in reference 26, which overcomes the singularity problem via the generalized singular value decomposition (GSVD)[27], also belongs to this family. Discriminant analysis with an orthogonal transformation has also been studied in reference 30.

1.3 A UNIFIED FRAMEWORK FOR GENERALIZED LDA

The LDA extensions discussed in the last section employ different techniques to deal with the singularity problem. In this section, we present a four-step general framework for various generalized LDA algorithms. The presented framework unifies most of the generalized LDA algorithms. The properties of various algorithms as well as their

relationships are elucidated from this framework. The unified framework consists of four steps described below:

1. Compute the eigenvalues, $\{\lambda_i\}_{i=1}^d$, of S_t in Eq. (1.3) and the corresponding eigenvectors $\{u_i\}_{i=1}^d$, with $\lambda_1 \geq \dots \geq \lambda_d$. Then S_t can be expressed as $S_t = \sum_{i=1}^d \lambda_i u_i u_i^T$.
2. Given a transfer function $\Phi : \mathbb{R} \rightarrow \mathbb{R}$, let $\tilde{\lambda}_i = \Phi(\lambda_i)$, for all i . Construct the matrix \tilde{S}_t as $\tilde{S}_t = \sum_{i=1}^d \tilde{\lambda}_i u_i u_i^T$.
3. Compute the eigenvectors, $\{\phi_i\}_{i=1}^q$, of $\tilde{S}_t^+ S_b$ corresponding to the nonzero eigenvalues, where $q = \text{rank}(S_b)$, \tilde{S}_t^+ denotes the pseudo-inverse of \tilde{S}_t [27]. Construct the matrix G as $G = [\phi_1, \dots, \phi_q]$.
4. Optional orthogonalization step: Compute the QR decomposition [27] of G as $G = QR$, where $Q \in \mathbb{R}^{d \times q}$ has orthonormal columns and $R \in \mathbb{R}^{q \times q}$ is upper triangular.

With this four-step procedure, the final transformation is given by either the matrix G from step 3, if the optional orthogonalization step is not applied, or the matrix Q from step 4 if the transformation matrix is required to be orthogonal. In this framework, different transfer functions, Φ , in step 2 lead to different generalized LDA algorithms, as summarized below:

- In PCA+LDA, the intermediate dimensionality reduction stage by PCA keeps the top p eigenvalues of S_t ; thus it applies the following linear step function: $\Phi(\lambda_i) = \lambda_i$, for $1 \leq i \leq p$, and $\Phi(\lambda_i) = 0$, for $i > p$. The optional orthogonalization step is not employed in PCA+LDA.
- In regularized LDA (RLDA), a regularization term is applied to S_t as $S_t + \mu I_d$, for some $\mu > 0$. It corresponds to the use of the following transfer function: $\Phi(\lambda_i) = \lambda_i + \mu$, for all i . The optional orthogonalization step is not employed in RLDA.
- In uncorrelated LDA (ULDA), the optimal transformation consists of the top eigenvectors of $S_t^+ S_b$ [24]. The corresponding transfer function is thus given by $\Phi(\lambda_i) = \lambda_i$, for all i . The same transfer function is used in orthogonal LDA (OLDA). The difference between ULDA and OLDA is that OLDA performs the optional orthogonalization step while it is not applied in ULDA.
- In orthogonal centroid method (OCM), the optimal transformation is given by the top eigenvectors of S_b [25]. The transfer function is thus given by $\Phi(\lambda_i) = 1$, for all i . Since the eigenvectors of S_b forms an orthonormal set, the optional orthogonalization step is not necessary in OCM.

It has been shown [31] that the regularization in RLDA is effective for nonzero eigenvalues only. Thus, we can apply the following transfer function for RLDA:

$$\Phi(\lambda_i) = \begin{cases} \lambda_i + \mu & \text{for } 1 \leq i \leq t, \\ 0 & \text{for } i > t, \end{cases}$$

Table 1.1. Transfer Functions for Different LDA Extensions

	PCA+LDA	RLDA	ULDA/OLDA	OCM
$\Phi(\lambda_i) =$	$\begin{cases} \lambda_i & \text{for } 1 \leq i \leq p \\ 0 & \text{for } i > p \end{cases}$	$\begin{cases} \lambda_i + \mu & \text{for } 1 \leq i \leq t \\ 0 & \text{for } i > t \end{cases}$	λ_i	1

where $t = \text{rank}(S_t)$. The transfer functions for different LDA extensions are summarized in Table 1.1.

In null space LDA (NLDA) [20, 32], the data are first projected onto the null space of S_w , which is then followed by classical LDA. It is not clear which transfer function Φ corresponds to the projection onto the null space of S_w . In reference 33, the equivalence relationship between NLDA and OLDA was established under a mild condition

$$C1 : \text{rank}(S_t) = \text{rank}(S_b) + \text{rank}(S_w), \quad (1.15)$$

which has been shown to hold for many high-dimensional data. Thus, for high-dimensional data, we can use the following transfer function for NLDA: $\Phi(\lambda_i) = \lambda_i$, for all i .

1.3.1 Analysis

The unified framework from the last section summarizes the commonalities and differences of various LDA-based algorithms. This unification of diverse algorithms into a common framework sheds light on the understanding of the key features of various algorithms as well as their relationship.

It is clear from Table 1.1 that ULDA is reduced to the OCM algorithm [25] when S_t is a multiple of the identity matrix. Recent studies on the geometric representation of high-dimensional and small sample size data show that under mild conditions, the covariance matrix S_t tends to a scaled identity matrix when the data dimension d tends to infinity with the sample size n fixed [34]. This implies that all the eigenvalues of S_t are the same. In other words, the data behave as if the underlying distribution is spherical. In this case, OCM is equivalent to ULDA. This partially explains the effectiveness of OCM when working on high-dimensional data.

We can observe from Table 1.1 that when the reduced dimensionality, p , in the PCA stage of PCA+LDA is chosen to be the rank of S_t —that is, the PCA stage keeps all the information—then the transfer functions for PCA+LDA and ULDA are identical. That is, PCA+LDA is equivalent to ULDA in this case. It can also be observed from Table 1.1 that the transfer function for RLDA equals the one for ULDA when $\mu = 0$. Thus, ULDA can be considered as a special case of both PCA+LDA and RLDA.

It follows from the above discussion that when $\mu = 0$ in RLDA, and $p = \text{rank}(S_t)$ in PCA+LDA, they both reduce to ULDA. It has been shown that, under condition C1 in Eq. (1.15), the transformation matrix of ULDA lies in the null space of S_w [33]. That is, $G^T S_w = 0$. Furthermore, it was shown in reference 31 that if $G^T S_w = 0$

holds, then the transformation matrix G maps all data points from the same class to a common vector. This is an extension of the result in reference 32, which assumes that all classes in the data set have the same number of samples. Thus it follows that the ULDA transformation maps all data points from the same class to a common vector, provided that condition C1 is satisfied. This leads to a perfect separation between different classes in the dimensionality-reduced space. However, it may also result in overfitting. RLDA overcomes this limitation by choosing a nonzero regularization value μ , while PCA+LDA overcomes this limitation by setting $p < \text{rank}(S_i)$.

The above analysis shows that the regularization in RLDA and the PCA dimensionality reduction in PCA+LDA are expected to alleviate the overfitting problem, provided that appropriate values for μ and p can be estimated. Selecting an optimal value for a parameter such as μ in RLDA and p in PCA+LDA from a given candidate set is called *model selection* [17]. Existing studies have focused on the estimation from a small candidate set, as it involves expensive matrix computations for each candidate value. However, a large candidate set is desirable in practice to achieve good performance. This has been one of the main reasons for their limited applicability in practice. To overcome this problem, an efficient model selection algorithm for RLDA was proposed in reference 31 and this algorithm can estimate an optimal value for μ from a large number of candidate values efficiently.

1.4 A LEAST SQUARES FORMULATION FOR LDA

In this section, we discuss recent developments on connecting LDA to multivariate linear regression (MLR). We first discuss the relationship between linear regression and LDA in the binary-class case. We then present multivariate linear regression with a specific class indicator matrix. This indicator matrix plays a key role in establishing the equivalence relationship between MLR and LDA in the multiclass case.

1.4.1 Linear Regression versus Fisher LDA

Given a data set of two classes, $\{(x_i, y_i)\}_{i=1}^n$, $x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, the linear regression model with the class label as the output has the following form:

$$f(x) = x^T w + b, \quad (1.16)$$

where $w \in \mathbb{R}^d$ is the weight vector, and b is the bias of the linear model. A popular approach for estimating w and b is to minimize the sum-of-squares error function, called least squares, as follows:

$$L(w, b) = \frac{1}{2} \sum_{i=1}^n \|f(x_i) - y_i\|^2 = \frac{1}{2} \|X^T w + be - y\|^2, \quad (1.17)$$

where $X = [x_1, x_2, \dots, x_n]$ is the data matrix, e is the vector of all ones, and y is the vector of class labels. Assume that both $\{x_i\}$ and $\{y_i\}$ have been centered, that is,

$\sum_{i=1}^n x_i = 0$ and $\sum_{i=1}^n y_i = 0$. It follows that

$$y_i \in \{-2n_2/n, 2n_1/n\},$$

where n_1 and n_2 denote the number of samples from the negative and positive classes, respectively. In this case, the bias term b in Eq. (1.16) becomes zero and we construct a linear model $f(x) = x^T w$ by minimizing

$$L(w) = \frac{1}{2} \|X^T w - y\|^2. \quad (1.18)$$

It can be shown that the optimal w minimizing the objective function in Eq. (1.18) is given by [16, 17]

$$w = (X X^T)^+ X y.$$

Note that the data matrix X has been centered and thus $X X^T = n S_t$ and $X y = \frac{2n_1 n_2}{n} (c^{(1)} - c^{(2)})$. It follows that

$$w = \frac{2n_1 n_2}{n^2} S_t^+ (c^{(1)} - c^{(2)}) = \frac{2n_1 n_2}{n^2} G^F,$$

where G^F is the optimal solution to FLDA in Eq. (1.14). Hence linear regression with the class label as the output is equivalent to Fisher LDA, as the projection in FLDA is invariant of scaling. More details on this equivalence relationship can be found in references 15, 16, and 35.

1.4.2 Relationship Between Multivariate Linear Regression and LDA

In the multiclass case, we are given a data set consisting of n samples $\{(x_i, y_i)\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ and $y_i \in \{1, 2, \dots, k\}$ denotes the class label of the i th sample and $k > 2$. To apply the least squares formalism to the multiclass case, the 1-of- k binary coding scheme is usually used to associate a vector-valued class code to each data point [15, 17]. In this coding scheme, the class indicator matrix, denoted as $Y_1 \in \mathbb{R}^{n \times k}$, is defined as follows:

$$Y_1(ij) = \begin{cases} 1 & \text{if } y_i = j, \\ 0 & \text{otherwise.} \end{cases} \quad (1.19)$$

It is known that the solution to least squares problem approximates the conditional expectation of the target values given the input [15]. One justification for using the 1-of- k scheme is that, under this coding scheme, the conditional expectation is given by the vector of posterior class probabilities. However, these probabilities are usually approximated rather poorly [15]. There are also some other class indicator matrices considered in the literature. In particular, the indicator matrix $Y_2 \in \mathbb{R}^{n \times k}$, defined as

$$Y_2(ij) = \begin{cases} 1 & \text{if } y_i = j, \\ -1/(k-1) & \text{otherwise,} \end{cases} \quad (1.20)$$

has been introduced to extend support vector machines (SVM) for multiclass classification [36] and to generalize the kernel target alignment measure [37], originally proposed in reference 38.

In multivariate linear regression, a k -tuple of discriminant functions

$$f(x) = (f_1(x), f_2(x), \dots, f_k(x))$$

is considered for each $x \in \mathbb{R}^d$. Denote $\tilde{X} = [\tilde{x}_1, \dots, \tilde{x}_n] \in \mathbb{R}^{d \times n}$ and $\tilde{Y} = (\tilde{Y}_{ij}) \in \mathbb{R}^{n \times k}$ as the centered data matrix X and the centered indicator matrix Y , respectively. That is, $\tilde{x}_i = x_i - \bar{x}$ and $\tilde{Y}_{ij} = Y_{ij} - \bar{Y}_j$, where $\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$ and $\bar{Y}_j = \frac{1}{n} \sum_{i=1}^n Y_{ij}$. Then MLR computes the weight vectors, $\{w_j\}_{j=1}^k \in \mathbb{R}^d$, of the k linear models, $f_j(x) = x^T w_j$, for $j = 1, \dots, k$, via the minimization of the following sum-of-squares error function:

$$L(W) = \frac{1}{2} \|\tilde{X}^T W - \tilde{Y}\|_F^2 = \frac{1}{2} \sum_{j=1}^k \sum_{i=1}^n \|f_j(\tilde{x}_i) - \tilde{Y}_{ij}\|^2, \quad (1.21)$$

where $W = [w_1, w_2, \dots, w_k]$ is the weight matrix and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix [27]. The optimal W is given by [15, 17]

$$W = (\tilde{X}\tilde{X}^T)^+ \tilde{X}\tilde{Y}, \quad (1.22)$$

which is dependent on the centered class indicator matrix \tilde{Y} .

Both Y_1 and Y_2 defined in Eqs. (1.19) and (1.20), as well as the one in reference 39, could be used to define the centered indicator matrix \tilde{Y} . An interesting connection between the linear regression model using Y_1 and LDA can be found in reference 17 (page 112). It can be shown that if $X^L = W_1^T \tilde{X}$ is the transformed data by W_1 , where $W_1 = (\tilde{X}\tilde{X}^T)^+ \tilde{X}\tilde{Y}_1$ is the least squares solution in Eq. (1.22) using the centered indicator matrix \tilde{Y}_1 , then LDA applied to X^L is identical to LDA applied to \tilde{X} in the original space. In this case, linear regression is applied as a preprocessing step before the classification and is in general not equivalent to LDA. The second indicator matrix Y_2 has been used in SVM, and the resulting model using Y_2 is also not equivalent to LDA in general. This is also the case for the indicator matrix in reference 39. One natural question is whether there exists a class indicator matrix $\tilde{Y} \in \mathbb{R}^{n \times k}$, with which multivariate linear regression is equivalent to LDA. If this is the case, then LDA can be formulated as a least squares problem in the multiclass case, and the generalizations of least squares can be readily applied to LDA.

In MLR, each \tilde{x}_i is transformed to

$$(f_1(\tilde{x}_i), \dots, f_k(\tilde{x}_i))^T = W^T \tilde{x}_i,$$

and the centered data matrix $\tilde{X} \in \mathbb{R}^{d \times n}$ is transformed to $W^T \tilde{X} \in \mathbb{R}^{k \times n}$, thus achieving dimensionality reduction if $k < d$. Note that the transformation matrix W in MLR is dependent on the centered class indicator matrix \tilde{Y} as in Eq. (1.22). To derive a class indicator matrix for MLR with which the transformation matrix is related to that of LDA, it is natural to apply the class discrimination criterion used in LDA. We thus

look for \tilde{Y} , which solves the following optimization problem:

$$\begin{aligned} \max_{\tilde{Y}} \quad & \text{trace} \left((W^T S_b W)(W^T S_t W)^+ \right) \\ \text{subject to} \quad & W = (\tilde{X} \tilde{X}^T)^+ \tilde{X} \tilde{Y}, \end{aligned} \quad (1.23)$$

where the pseudo-inverse is used as the matrix $\tilde{X} \tilde{X}^T$ can be singular.

In reference 40, a new class indicator matrix, called Y_3 , is constructed and it was shown that Y_3 solves the optimization problem in Eq. (1.23). This new class indicator matrix $Y_3 = (Y_3(ij))_{ij} \in \mathbb{R}^{n \times k}$ is defined as follows:

$$Y_3(ij) = \begin{cases} \sqrt{\frac{n}{n_j}} - \sqrt{\frac{n_j}{n}} & \text{if } y_i = j, \\ -\sqrt{\frac{n_j}{n}} & \text{otherwise,} \end{cases} \quad (1.24)$$

where n_j is the sample size of the j th class, and n is the total sample size. Note that Y_3 defined above has been centered (in terms of rows), and thus $\tilde{Y}_3 = Y_3$. More importantly, it was shown in reference 40 that, under condition C1 in Eq. (1.15), multivariate linear regression with Y_3 as the class indicator matrix is equivalent to LDA. We outline the main result below and the detailed proof can be found in reference 40.

Recall that in LDA, the optimal transformation matrix (G^{LDA}) consists of the top eigenvectors of $S_t^+ S_b$ corresponding to the nonzero eigenvalues. On the other hand, since $\tilde{X} \tilde{X}^T = n S_t$ and $\tilde{X} Y_3 = n H_b$, where S_t and H_b are defined in Eqs. (1.3) and (1.5), respectively, the optimal weight matrix W^{MLR} for MLR in Eq. (1.22) can be expressed as

$$W^{MLR} = (\tilde{X} \tilde{X}^T)^+ \tilde{X} Y_3 = (n S_t)^+ n H_b = S_t^+ H_b. \quad (1.25)$$

It can be shown that the transformation matrix G^{LDA} of LDA, which consists of the top eigenvectors of $S_t^+ S_b$, and the projection matrix for MLR that is given in Eq. (1.25) are related as follows [40]:

$$W^{MLR} = \left[G^{LDA} \Sigma, 0 \right] Q^T,$$

where Σ is a diagonal matrix and Q is an orthogonal matrix.

The K-Nearest-Neighbor (K-NN) algorithm [16] based on the Euclidean distance is commonly applied as the classifier in the dimensionality-reduced space of LDA. If we apply W^{MLR} for dimensionality reduction before K-NN, the matrix W^{MLR} is invariant of an orthogonal transformation, since any orthogonal transformation preserves all pairwise distance. Thus W^{MLR} is essentially equivalent to $\left[G^{LDA} \Sigma, 0 \right]$ or $G^{LDA} \Sigma$, as the removal of zero columns does not change the pairwise distance either. Thus the essential difference between W^{MLR} and G^{LDA} is the diagonal matrix Σ . Interestingly, it was shown in reference 40 that the matrix Σ is an identity matrix under the condition C1 defined in Eq. (1.15). This implies that multivariate linear regression with Y_3 as the class indicator matrix is equivalent to LDA provided that

the condition C1 is satisfied. Thus LDA can be formulated as a least squares problem in the multiclass case. Experimental results in reference 40 show that condition C1 is likely to hold for high-dimensional and undersampled data.

1.5 SEMISUPERVISED LDA

Semisupervised learning, which occupies the middle ground between supervised learning (in which all training examples are labeled) and unsupervised learning (in which no labeled data are given), has received considerable attention recently [41–43]. The least square LDA formulation from the last section results in Laplacian-regularized LDA [44]. Furthermore, it naturally leads to semisupervised dimensionality reduction by incorporating the unlabeled data through the graph Laplacian.

1.5.1 Graph Laplacian

Given a data set $\{x_i\}_{i=1}^n$, a weighted graph can be constructed where each node in the graph corresponds to a data point in the data set. The weight S_{ij} between two nodes x_i and x_j is commonly defined as follows:

$$S_{ij} = \begin{cases} \exp\left(-\frac{\|x_i - x_j\|^2}{\sigma}\right), & x_i \in N_\kappa(x_j) \text{ or } x_j \in N_\kappa(x_i), \\ 0 & \text{otherwise,} \end{cases} \quad (1.26)$$

where both κ and $\sigma > 0$ are parameters to be specified, and $x_i \in N_\kappa(x_j)$ implies that x_i is among the κ nearest neighbors of x_j [45]. Let S be the similarity matrix whose (i, j) th entry is S_{ij} . To learn an appropriate representation $\{z_i\}_{i=1}^n$ which preserves locality structure, it is common to minimize the following objective function [45]:

$$\sum_{i,j} \|z_i - z_j\|^2 S_{ij}. \quad (1.27)$$

Intuitively, if x_i and x_j are close to each other in the original space—that is, S_{ij} is large—then $\|z_i - z_j\|$ tends to be small if the objective function in Eq. (1.27) is minimized. Thus the locality structure in the original space is preserved.

Define the Laplacian matrix L as $L = D - S$, where D is a diagonal matrix whose diagonal entries are the column sums of S . That is, $D_{ii} = \sum_{j=1}^n S_{ij}$. Note that L is symmetric and positive semidefinite. It can be verified that

$$\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \|z_i - z_j\|^2 S_{ij} = \text{trace}(ZLZ^T), \quad (1.28)$$

where $Z = [z_1, \dots, z_n]$.

1.5.2 A Regularization Framework for Semisupervised LDA

In semisupervised LDA, information from unlabeled data is incorporated into the formulation via a regularization term defined as in Eq. (1.28). Mathematically, semisupervised LDA computes an optimal weight matrix W^* , which solves the following optimization problem:

$$W^* = \arg \min_W \left\{ \|\tilde{X}^T W - Y_3\|_F^2 + \gamma \text{trace}(W^T \tilde{X} L \tilde{X}^T W) \right\}, \quad (1.29)$$

where $\gamma \geq 0$ is a tuning parameter and Y_3 is the class indicator matrix defined in Eq. (1.24). Since the Laplacian regularizer in Eq. (1.29) does not depend on the label information, the unlabeled data can be readily incorporated into the formulation. Thus the locality structures of both labeled and unlabeled data points are captured through the transformation W . It is clear that W^* is given by

$$W^* = (\gamma \tilde{X} L \tilde{X}^T + \tilde{X} \tilde{X}^T)^+ {}_n H_b. \quad (1.30)$$

1.6 EXTENSIONS TO KERNEL-INDUCED FEATURE SPACE

The discussion so far focuses on linear dimensionality reduction and regression. It has been shown that both discriminant analysis and regression can be adapted to nonlinear models by using the kernel trick [46–48]. Mika et al. [49] extended the Fisher discriminant analysis to its kernel version in the binary-class case. Following the work in reference 50, Baudat and Anouar [51] proposed the generalized discriminant analysis (GDA) algorithm for multiclass problems. The equivalence relationship between kernel discriminant analysis (KDA) and kernel regression has been studied in reference 35 for binary-class problems. The analysis presented in this chapter can be applied to extend this equivalence result to multiclass problems.

A symmetric function $\kappa : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, where \mathcal{X} denotes the input space, is called a kernel function if it satisfies the finitely positive semidefinite property [46]. That is, for any $x_1, \dots, x_n \in \mathcal{X}$, the kernel Gram matrix $K \in \mathbb{R}^{n \times n}$, defined by $K_{ij} = \kappa(x_i, x_j)$, is positive semidefinite. Any kernel function κ implicitly maps the input set \mathcal{X} to a high-dimensional (possibly infinite) Hilbert space \mathcal{H}_κ equipped with the inner product $(\cdot, \cdot)_{\mathcal{H}_\kappa}$ through a mapping ϕ_κ from \mathcal{X} to \mathcal{H}_κ :

$$\kappa(x, z) = (\phi_\kappa(x), \phi_\kappa(z))_{\mathcal{H}_\kappa}.$$

In KDA, three scatter matrices are defined in the feature space \mathcal{H}_κ as follows:

$$S_w^\phi = \frac{1}{n} \sum_{j=1}^k \sum_{x \in X_j} (\phi(x) - c_j^\phi) (\phi(x) - c_j^\phi)^T, \quad (1.31)$$

$$S_b^\phi = \frac{1}{n} \sum_{j=1}^k n_j \left(c_j^\phi - c^\phi \right) \left(c_j^\phi - c^\phi \right)^T, \quad (1.32)$$

$$S_t^\phi = \frac{1}{n} \sum_{j=1}^k \sum_{x \in X_j} \left(\phi(x) - c^\phi \right) \left(\phi(x) - c^\phi \right)^T, \quad (1.33)$$

where c_j^ϕ is the centroid of the j th class and c^ϕ is the global centroid in the feature space. Similar to the linear case, the transformation \mathcal{G} of KDA can be computed by solving the following optimization problem:

$$\mathcal{G} = \arg \max_{\mathcal{G}} \left\{ \text{trace} \left(\left(\mathcal{G}^T S_t^\phi \mathcal{G} \right)^+ \mathcal{G}^T S_b^\phi \mathcal{G} \right) \right\}. \quad (1.34)$$

It follows from the Representer Theorem [47] that columns of \mathcal{G} lie in the span of the images of training data in the feature space. That is,

$$\mathcal{G} = \phi(X)B, \quad (1.35)$$

for some matrix $B \in \mathbb{R}^{n \times (k-1)}$, where

$$\phi(X) = [\phi(x_1), \dots, \phi(x_n)]$$

is the data matrix in the feature space. Substituting Eq. (1.35) into Eq. (1.34), we can obtain the matrix B by solving the following optimization problem:

$$B = \arg \max_B \left\{ \text{trace} \left(\left(B^T S_t^K B \right)^+ B^T S_b^K B \right) \right\}, \quad (1.36)$$

where $S_b^K = KY_3Y_3^TK$, $S_t^K = K^2$, and $K = \phi(X)^T\phi(X)$ is the kernel matrix.

It can be verified that S_b^K and S_t^K are the between-class and total scatter matrices, respectively, when each column in K is considered as a data point in the n -dimensional space. It follows from Theorem 5.3 in reference 33 that the condition C1 in Eq. (1.15) is satisfied if all the training data points are linearly independent. Therefore, if the kernel matrix K is nonsingular (hence its columns are linearly independent), then kernel discriminant analysis (KDA) and kernel regression using Y_3 as the class indicator matrix are essentially equivalent. This extends the equivalence result between KDA and kernel regression in the binary-class case, originally proposed in reference 35, to the multiclass setting.

To overcome the singularity problem in kernel discriminant analysis (KDA), a number of techniques have been developed in the literature. Regularization was employed in reference 52. The QR decomposition was employed in reference 51 to avoid the singularity problem by removing the zero eigenvalues. Lu et al. [53, 54] extended the direct LDA (DLDA) algorithm [55] to kernel direct LDA based on the kernel trick. PCA+LDA was discussed in reference 56, and a *complete* algorithm was proposed to derive discriminant vectors from the null space of the within-class scatter matrix and its orthogonal complement. Recently, similar ideas were extended to the feature space based on kernel PCA [57].

Another challenging issue in applying KDA is the selection of an appropriate kernel function. Recall that kernel methods work by embedding the input data into some high-dimensional feature space. The key fact underlying the success of kernel methods is that the embedding into feature space can be determined uniquely by specifying a kernel function that computes the dot product between data points in the feature space. In other words, the kernel function implicitly defines the nonlinear mapping to the feature space and expensive computations in the high-dimensional feature space can be avoided by evaluating the kernel function. Thus one of the central issues in kernel methods is the selection of kernels.

To automate kernel-based learning algorithms, it is desirable to integrate the tuning of kernels into the learning process. This problem has been addressed from different perspectives recently. Lanckriet et al. [58] pioneered the work of multiple kernel learning (MKL) in which the optimal kernel matrix is obtained as a linear combination of prespecified kernel matrices. It was shown [58] that the coefficients in MKL can be determined by solving convex programs in the case of Support Vector Machines (SVM). While most existing work focuses on learning kernels for SVM, Fung et al. [59] proposed to learn kernels for discriminant analysis. Based on ideas from MKL, this problem was reformulated as a semidefinite program (SDP) [60] in reference 61 for binary-class problems.

By optimizing an alternative criterion, an SDP formulation for the KDA kernel learning problem in the multiclass case was proposed in reference 62. To reduce the computational cost of the SDP formulation, an approximate scheme was also developed. Furthermore, it was shown that the regularization parameter for KDA can also be learned automatically in this framework [62]. Although the approximate SDP formulation in reference 62 is scalable in terms of the number of classes, interior point algorithms [63] for solving SDP have an inherently large time complexity and thus it can not be applied to large-scale problems. To improve the efficiency of this formulation, a quadratically constrained quadratic program (QCQP) [63] formulation was proposed in reference 64 and it is more scalable than the SDP formulations.

1.7 OTHER LDA EXTENSIONS

Sparsity has recently received much attention for extending existing algorithms to induce sparse solutions [65–67]. L_1 -norm penalty has been used in regression [68], known as LASSO, and SVM [69, 70] to achieve model sparsity. Sparsity often leads to easy interpretation and good generalization ability of the resulting model. Sparse Fisher LDA has been proposed in reference 35, for binary-class problems. Based on the equivalence relationship between LDA and MLR, a multiclass sparse LDA formulation was proposed in reference 71 and an entire solution path for LDA was also obtained through the LARS algorithm [72].

The discussions in this chapter focus on supervised approaches. In the unsupervised setting, LDA can be applied to find the discriminant subspace for clustering, such as K-means clustering. In this case, an iterative algorithm can be derived alternating between clustering and discriminant subspace learning via LDA [73–75].

Interestingly, it can be shown that this iterative procedure can be simplified and is essentially equivalent to kernel K-means with a specific kernel Gram matrix [76].

When the data in question are given as high-order representations such as 2D and 3D images, it is natural to encode them using high-order tensors. Discriminant tensor factorization, which is a two-dimensional extension of LDA, for a collection of two-dimensional images has been studied [77]. It was further extended to higher-order tensors in reference 78. However, the computational convergence of these iterative algorithms [77, 78] is not guaranteed. Recently, a novel discriminant tensor factorization procedure with the convergence property was proposed [79]. Other recent extensions on discriminant tensor factorization as well as their applications to image analysis can be found in reference 80.

1.8 CONCLUSION

In this chapter, we provide a unified view of various LDA algorithms and discuss recent developments on connecting LDA to multivariate linear regression. We show that MLR with a specific class indicator matrix is equivalent to LDA under a mild condition, which has been shown to hold for many high-dimensional and small sample size data. This implies that LDA reduces to a least squares problem under this condition, and its solution can be obtained by solving a system of linear equations. Based on this equivalence result, we show that LDA can be applied in the semisupervised setting. We further extend the discussion to the kernel-induced feature space and present recent developments on kernel learning. Finally, we discuss several other recent developments on discriminant analysis, including sparse LDA, unsupervised LDA, and tensor LDA.

REFERENCES

1. A. K. Jain, P. Flynn, and A. A. Ross, *Handbook of Biometrics*, Springer, New York, 2007.
2. A. K. Jain and S. Z. Li, *Handbook of Face Recognition*, Springer-Verlag, New York, 2005.
3. A. K. Jain, A. A. Ross, and S. Prabhakar, An introduction to biometric recognition, *IEEE Trans. Circuits Syst. Video Technol.* **14**(1):4–20, 2004.
4. R. E. Bellman, *Adaptive Control Processes: A Guided Tour*, Princeton University Press, Princeton, NJ, 1961.
5. D. Donoho, High-dimensional data analysis: The curses and blessings of dimensionality, in *American Mathematical Society Lecture—Math Challenges of the 21st Century*, August 2000.
6. S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(1):40–51, 2007.
7. R. A. Fisher, The use of multiple measurements in taxonomic problems. *Ann. Eugenics* **7**:179–188, 1936.
8. K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd edition, Academic Press Professional, San Diego, 1990.
9. I. T. Jolliffe, *Principal Component Analysis*, 2nd edition, Springer-Verlag, New York, 2002.
10. M. Belkin and P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, *Neural Comput.* **15**(6):1373–1396, 2003.

11. C. J. C. Burges. Geometric methods for feature extraction and dimensional reduction, in O. Maimon and L. Rokach, editors, *Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*, Springer, New York, 2005, pp. 59–92.
12. S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding, *Science* **290**(5500):2323–6, 2000.
13. L. K. Saul, K. Q. Weinberger, J. H. Ham, F. Sha, and D. D. Lee. Spectral methods for dimensionality reduction, in O. Chapelle B. Schölkopf and A. Zien, editors, *Semisupervised Learning*, MIT Press, Cambridge, MA, 293–308, 2006.
14. J. B. Tenenbaum, V. d. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction, *Science* **290**(5500):279–294, 2000.
15. C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, New York, 2006.
16. R. O. Duda, P. E. Hart, and D. Stork, *Pattern Classification*, John Wiley & Sons, New York, 2000.
17. T. Hastie, R. Tibshirani, and J.H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, 2001.
18. A. M. Martinez and M. Zhu. Where are linear feature extraction methods applicable? *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(12):1934–1944, 2005.
19. P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.* **19**(7):711–720, 1997.
20. 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 Recognit* **33**:1713–1726, 2000.
21. Y. Guo, T. Hastie, and R. Tibshirani. Regularized linear discriminant analysis and its application in microarrays, *Biostatistics* **8**(1):86–100, 2007.
22. A. M. Martinez and A. C. Kak. PCA versus LDA, *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(2):228–233, 2001.
23. X. Wang and X. Tang. A unified framework for subspace face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(9):1222–1228, 2004.
24. J. Ye. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems, *J. Mach. Learning Res.* **6**:483–502, 2005.
25. H. Park, M. Jeon, and J. B. Rosen. Lower dimensional representation of text data based on centroids and least squares, *BIT* **43**(2):1–22, 2003.
26. P. Howland, M. Jeon, and H. Park. Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition, *SIAM J. Matrix Anal. Appl.* **25**(1):165–179, 2003.
27. G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd edition, The Johns Hopkins University Press, Baltimore, 1996.
28. W. Zhao, R. Chellappa, and P. Phillips. Subspace linear discriminant analysis for face recognition, Technical Report CAR-TR-914, Center for Automation Research, University of Maryland, 1999.
29. V. N. Vapnik. *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.
30. L. Duchene and S. Leclercq. An optimal transformation for discriminant and principal component analysis, *IEEE Trans. Pattern Anal. Mach. Intell.* **10**(6):978–983, 1988.
31. J. Ye, T. Xiong, Q. Li, R. Janardan, J. Bi, V. Cherkassky, and C. Kambhampettu. Efficient model selection for regularized linear discriminant analysis, in *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, 2006, pp. 532–539.
32. H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana. Discriminative common vectors for face recognition, *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(1):4–13, 2005.
33. J. Ye and T. Xiong. Computational and theoretical analysis of null space and orthogonal linear discriminant analysis, *J. Mach. Learning Res.* **7**:1183–1204, 2006.
34. P. Hall, J. S. Marron, and A. Neeman. Geometric representation of high dimension, low sample size data, *J. R. Stat. Soc. Ser. B* **67**:427–444, 2005.
35. S. Mika, *Kernel Fisher Discriminants*, Ph.D. thesis, University of Technology, Berlin, 2002.
36. Y. Lee, Y. Lin, and G. Wahba. Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data, *J. Am. Stat. Assoc.* **99**:67–81, 2004.

37. Y. Guermeur, A. Lifchitz, and R. Vert, A kernel for protein secondary structure prediction, in *Kernel Methods in Computational Biology*, The MIT Press, Cambridge, MA, 2004, pp. 193–206.
38. N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor, On kernel target alignment, in *Advances in Neural Information Processing Systems*, The MIT Press, Cambridge, MA, 2001.
39. C. Park and H. Park, A relationship between LDA and the generalized minimum squared error solution, *SIAM J. Matrix Anal. Appl.* **27**(2):474–492, 2005.
40. J. Ye, Least squares linear discriminant analysis, in *Proceedings of the 24th International Conference on Machine Learning*, 2007, pp. 1087–1093.
41. O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.
42. D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, Learning with local and global consistency, in *Advances in Neural Information Processing Systems*, 2003, pp. 321–328.
43. X. Zhu, Z. Ghahramani, and J. Lafferty, Semi-supervised learning using Gaussian fields and harmonic functions, in *Proceedings of the 20th International Conference on Machine Learning*, 2003, pp. 912–919.
44. J. Chen, J. Ye, and Q. Li, Integrating global and local structures: A least squares framework for dimensionality reduction, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1–8.
45. M. Belkin and P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, *Adv. Neural Inf. Processing Sys.* **15**:585–591, 2001.
46. N. Cristianini and J. S. Taylor, *An Introduction to Support Vector Machines and other Kernel-Based Learning Methods*, Cambridge University Press, New York, 2000.
47. S. Schölkopf and A. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, Cambridge, MA, 2002.
48. J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, New York, 2004.
49. S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller, Fisher discriminant analysis with kernels, in Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, IEEE, New York, 1999, pp. 41–48.
50. B. Schölkopf, A. J. Smola, and K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, *Neural Comput.* **10**(5):1299–1319, 1998.
51. G. Baudat and F. Anouar, Generalized discriminant analysis using a kernel approach, *Neural Comput.* **12**(10):2385–2404, 2000.
52. S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K.-R. Müller, Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces, *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5):623–633, 2003.
53. J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, Face recognition using kernel direct discriminant analysis algorithms, *IEEE Trans. Neural Networks* **14**(1):117–126, 2003.
54. J. Lu, K. N. Plataniotis, A. N. Venetsanopoulos, and J. Wang, An efficient kernel discriminant analysis method, *Pattern Recognit.* **38**(10):1788–1790, 2005.
55. H. Yu and J. Yang, A direct LDA algorithm for high-dimensional data with applications to face recognition, *Pattern Recognit.* **34**:2067–2070, 2001.
56. J. Yang and J. Yang, Why can LDA be performed in PCA transformed space? *Pattern Recognit.* **36**(2):563–566, 2003.
57. J. Yang, A. F. Frangi, J. Yang, D. Zhang, and Z. Jin, KPCA plus LDA: A complete kernel fisher discriminant framework for feature extraction and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(2):230–244, 2005.
58. G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. E. Ghaoui, and M. I. Jordan, Learning the kernel matrix with semidefinite programming, *J. Mach. Learning Res.* **5**:27–72, 2004.
59. G. Fung, M. Dundar, J. Bi, and B. Rao, A fast iterative algorithm for Fisher discriminant using heterogeneous kernels, in *Proceedings of the Twenty-First International Conference on Machine Learning*, 2004.
60. L. Vandenberghe and S. Boyd, Semidefinite programming, *SIAM Rev.* **38**(1):49–95, 1996.

61. S.-J. Kim, A. Magnani, and S. Boyd, Optimal kernel selection in kernel Fisher discriminant analysis, in *Proceedings of the Twenty-Third International Conference on Machine Learning*, 2006, pp. 465–472.
62. J. Ye, J. Chen, and S. Ji, Discriminant kernel and regularization parameter learning via semidefinite programming, in *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, 2007, pp. 1095–1102.
63. S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, New York, 2004.
64. J. Ye, S. Ji, and J. Chen, Learning the kernel matrix in discriminant analysis via quadratically constrained quadratic programming, in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007, pp. 854–863.
65. A. d’Aspremont, L. E. Ghaoui, M. I. Jordan, and G. R. G. Lanckriet, A direct formulation for sparse PCA using semidefinite programming, *SIAM Rev.* **49**(3):434–448, 2007.
66. I. T. Jolliffe and M. Uddin, A modified principal component technique based on the lasso, *J. Comput. Graph. Stat.* **12**:531–547, 2003.
67. H. Zou, T. Hastie, and R. Tibshirani, Sparse principal component analysis, *J. Comput. Graph. Stat.* **15**(2):265–286, 2006.
68. R. Tibshirani, Regression shrinkage and selection via the lasso, *J. Royal Stat. Soc. Ser. B* (1):267–288, 1996.
69. L. Wang and X. Shen, On L_1 -norm multiclass support vector machines: Methodology and theory, *J. Am. Stat. Assoc.* **102**(478):583–594, 2007.
70. J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, 1-Norm support vector machines, in *Advances in Neural Information Processing Systems*, 2003.
71. J. Ye, J. Chen, R. Janardan, and S. Kumar, Developmental stage annotation of *Drosophila* gene expression pattern images via an entire solution path for LDA, in *ACM Transactions on Knowledge Discovery from Data, Special Issue on Bioinformatics*, 2008.
72. B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, Least angle regression (with discussion). *Ann. Stat.*, **32**(2):407–499, 2004.
73. F. De la Torre and T. Kanade, Discriminative cluster analysis, in *Proceedings of the Twenty-Third International Conference on Machine Learning*, 2006, pp. 241–248.
74. C. Ding and T. Li, Adaptive dimension reduction using discriminant analysis and k-means clustering, in *Proceedings of the Twenty-Fourth International Conference on Machine Learning*, 521–528, 2007.
75. J. Ye, Z. Zhao, and H. Liu, Adaptive distance metric learning for clustering, in *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
76. J. Ye, Z. Zhao, and M. Wu, Discriminative k -means for clustering, in *Proceedings of the Annual Conference on Advances in Neural Information Processing Systems*, 2007, pp. 1649–1656.
77. J. Ye, R. Janardan, and Q. Li, Two-dimensional linear discriminant analysis, in *Proceedings of the Annual Conference on Advances in Neural Information Processing Systems*, 2004, pp. 1569–1576.
78. S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H. Zhang, Discriminant analysis with tensor representation, in *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, 2005.
79. H. Wang, S. Yan, T. Huang, and X. Tang, A convergent solution to tensor subspace learning, in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2007.
80. D. Tao, X. Li, X. Wu, and S. J. Maybank, General tensor discriminant analysis and gabor features for gait recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(10):1700–1715, 2007.

