

CPM: A Covariance-preserving Projection Method

Jieping Ye*

Tao Xiong[†]

Ravi Janardan[‡]

Abstract

Dimension reduction is critical in many areas of data mining and machine learning. In this paper, a Covariance-preserving Projection Method (CPM for short) is proposed for dimension reduction. CPM maximizes the class discrimination and also preserves approximately the class covariance. The optimization involved in CPM can be formulated as low rank approximations of a collection of matrices, which can be solved iteratively. Our theoretical and empirical analysis reveals the relationship between CPM and Linear Discriminant Analysis (LDA), Sliced Average Variance Estimator (SAVE), and Heteroscedastic Discriminant Analysis (HDA). This gives us new insights into the nature of these different algorithms. We use both synthetic and real-world datasets to evaluate the effectiveness of the proposed algorithm.

keywords: Dimension reduction, linear discriminant analysis, heteroscedastic discriminant analysis, covariance.

1 Introduction

Linear Discriminant Analysis (LDA) [4, 7, 8] is a well-known scheme for feature extraction and dimension reduction. LDA projects the data onto a lower-dimensional vector space such that the ratio of the between-class distance to the within-class distance is maximized, thus achieving maximum discrimination. LDA is equivalent to maximum likelihood classification assuming normal distribution for each class with a common covariance matrix. It has been applied successfully to many areas, such as computer vision, bioinformatics, etc. [1, 6, 11, 15] However, LDA has several limitations:

- (1) It fails to recover the features for classification, when all class centroids coincide;

- (2) it may not find the best projection, when class covariance matrices vary; and
- (3) the reduced dimension of LDA is no larger than $k - 1$ (k denotes the number of classes), which may not be sufficient for complex data.

Generalization of LDA by fitting Gaussian mixtures to each class has been studied by Hastie [10]. Cook *et al.* [2, 3] proposed the Sliced Average Variance Estimator (SAVE), which is shown to be capable of dealing with the limitations in LDA. Zhu and Hastie [16] developed a general method for finding important discriminant directions without assuming the class densities belong to any particular parametric family. Kumar and Andreou [13] proposed HDA, based on a different model, in which the classes are still Gaussian, yet are allowed to have different covariance matrices, under the condition that both centroids and covariance matrices coincide in a subspace of the observation space.

In this paper, we propose a new algorithm for dimension reduction, called CPM (which stands for Covariance-preserving Projection Method). CPM aims to maximize the class discrimination and at the same time preserve approximately the class covariance by applying a tuning parameter α between 0 and 1. One key feature of CPM is that the critical information on the class covariance is preserved under the projection. With a properly chosen tuning parameter α , which may be dependent on the data distribution, the CPM algorithm is able to deal with difficult situations encountered in LDA. In practice, the best value of α can be estimated by cross-validation.

To illustrate the difference between CPM and LDA, we generated a synthetic dataset with 20 dimensions and 3 classes as in [16]. Fig 1 (top) shows the first two coordinates. For the first two coordinates, class 1 is simulated from a standard multivariate Gaussian, while classes 2 and 3 are mixtures of two symmetrically shifted standard Gaussians. In the remaining 18 coordinates, Gaussian noise with zero mean and standard deviation 1 is used for all three classes. It is clear from Fig 1 (middle) that LDA fails to extract important features because the class centroids coincide. CPM separates the three classes completely as shown in Fig 1 (bottom), which shows the advantage of incorporating the class

*Department of Computer Science and Engineering, Arizona State University

[†]Department of Electrical and Computer Engineering, University of Minnesota

[‡]Department of Computer Science and Engineering, University of Minnesota

covariance information in CPM.

The optimization (maximization) problem involved in CPM is nonlinear and difficult to solve. We formulate a lower bound for the criterion function used in CPM. The maximization of the derived lower bound can be formulated as low rank approximations of a collection of symmetric and positive semi-definite matrices, a special case of the low rank approximations in [5, 14]. To our best knowledge, there is no closed form solution. We derive an iterative algorithm, which updates the projection successively and converges to a local optimum. Unlike LDA, the reduced dimension, d , of CPM can be larger than $k - 1$.

We also study the relationship between CPM and SAVE and HDA. SAVE is shown to be closely related to a special case of CPM when $\alpha = 0.5$. Recall that the parameter α controls the tradeoff between the separation of class centroids and the preservation of class covariances. The optimal value of α may depend on the data distribution. CPM is more flexible in dealing with different situations by varying the values of α , than SAVE, where α is fixed to be 0.5.

Our theoretical analysis also reveals that CPM is an approximation of HDA. Note that HDA involves complex and nonlinear optimizations [13]. It has been observed in [13] that HDA has high computational cost, especially for large datasets; and HDA may not be complete, since the numerical optimization procedure in HDA may not converge. We show that the optimization problem involved in CPM is simpler and much easier to solve than HDA.

The theoretical analysis further justifies the use of the covariance information in CPM and gives us new insights into the nature of these different algorithms.

We perform experiments on both synthetic and real-world datasets to evaluate the effectiveness of CPM and compare it with other well known algorithms. Our experiments show that:

- (1) CPM is able to recover the features for classification, even when all class centroids coincide or the covariance matrices vary;
- (2) CPM is competitive with SAVE and LDA in classification;
- (3) CPM is comparable to HDA in classification; and
- (4) HDA does not complete for several cases.

Thus CPM may be a good alternative for HDA as a generalization of LDA to deal with difficult situations where all class centroids coincide or the covariance matrices vary.

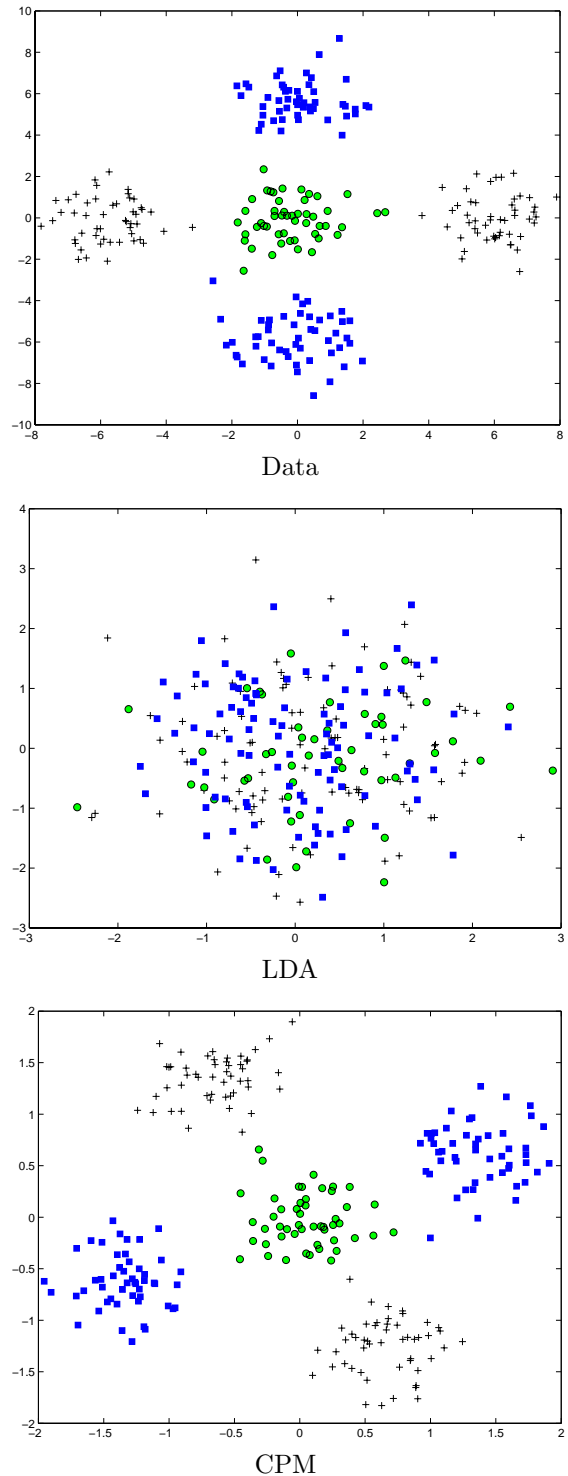


Figure 1: Top: original data (the first two coordinates); Middle: projection by LDA; and Bottom: projection by CPM. Note that the scales for different algorithms vary.

The rest of the paper is organized as follows: Section 2 presents the CPM algorithm; the relationship between CPM and SAVE and HDA is discussed in Section 3; experiments results are presented in Section 4; and the paper is concluded in Section 5.

2 Class Covariance-preserving Projection Method

Given a data matrix $A \in \mathbb{R}^{n \times N}$, consisting of N data points in n -dimensional space, we wish to find a vector space \mathcal{G} spanned by $\{g_i\}_{i=1}^d$, where $g_i \in \mathbb{R}^n$, such that each data point $a_i \in \mathbb{R}^n$ of A , is projected onto \mathcal{G} by

$$G^T a_i = (g_1^T \cdot a_i, \dots, g_d^T \cdot a_i)^T \in \mathbb{R}^d,$$

with $d < n$. Here

$$G = [g_1, \dots, g_d] \in \mathbb{R}^{n \times d}$$

denotes the projection.

Assume that the data in A is partitioned into k classes as $A = \{\Pi_1, \dots, \Pi_k\}$, where Π_i contains N_i data points from the i -th class, and $\sum_{i=1}^k N_i = N$. The covariance matrix of the i -th class is defined as

$$W_i = \frac{1}{N_i} \sum_{x \in \Pi_i} (x - c_i)(x - c_i)^T,$$

where

$$c_i = \frac{1}{N_i} \sum_{x \in \Pi_i} x$$

is the *centroid* of the i -th class,

In discriminant analysis, three scatter matrices, called *within-class* (S_w), *between-class* (S_b), and *total* (S_t) matrices are defined as follows [7]:

$$\begin{aligned} S_w &= \frac{1}{N} \sum_{i=1}^k N_i W_i, \\ S_b &= \frac{1}{N} \sum_{i=1}^k N_i (c_i - c)(c_i - c)^T, \\ S_t &= \frac{1}{N} \sum_{i=1}^k \sum_{x \in \Pi_i} (x - c)(x - c)^T, \end{aligned}$$

where

$$c = \frac{1}{N} \sum_{i=1}^k \sum_{x \in \Pi_i} x$$

is the *global centroid*. It is easy to verify that

$$S_t = S_w + S_b.$$

In the low-dimensional space resulting from the linear projection G , the scatter matrices become

$$S_b^L = G^T S_b G, \quad S_w^L = G^T S_w G, \quad \text{and} \quad S_t^L = G^T S_t G.$$

The optimal projection G^* in LDA [7] can be computed by solving the following optimization problem:

$$(2.1) \quad G^* = \arg \max_{G: G^T S_t G = I_d} \{\text{trace}(G^T S_b G)\}.$$

Assuming that the total scatter matrix has been normalized, i.e., $S_t = I_n$, the optimal projection for LDA can be obtained by maximizing $\text{trace}(G^T S_b G)$, subject to the orthogonality constraint, that is, $G^T G = I_d$. The solution is given by the top eigenvectors of S_b . There are at most $k - 1$ eigenvectors corresponding to the nonzero eigenvalues, since the rank of the matrix S_b is bounded from above by $k - 1$. Therefore, the reduced dimension of LDA is at most $k - 1$.

2.1 Problem formulation LDA maximizes the separation between different classes by maximizing $\text{trace}(G^T S_b G)$. It is optimal when all classes have a common covariance matrix. However, it ignores the class covariance information and may not be effective when the class centroids are close to each other (Note that $S_b = 0$ when class centroids coincide) or the class covariance matrices vary. The CPM algorithm proposed below attempts to overcome the limitations of LDA, by considering the information from both the class centroids and the class covariances.

Let us first consider Principal Component Analysis (PCA) [12] on the i -th class. Let G be the optimal projection consisting of the first d principal components of W_i (the covariance matrix of the i -th class). The information loss, or the approximation error by keep only the d principal components is

$$\|W_i - G(G^T W_i G)G^T\|_F,$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix [9]. The projection in PCA is shown to minimize the approximation error [12] and can be computed via the Singular Value Decomposition (SVD) [9]. In CPM, we consider all k classes simultaneously and aim to find the projection G such that the weighted approximation error of all classes is minimized. Mathematically, CPM aims to minimize the following weighted approximation error:

$$(2.2) \quad \sum_{i=1}^k \frac{N_i}{N} \|(W_i - S_w) - G(G^T(W_i - S_w)G)G^T\|_F^2,$$

which can be shown to be equal to

$$\sum_{i=1}^k \frac{N_i}{N} \|W_i - S_w\|_F^2 - \sum_{i=1}^k \frac{N_i}{N} \|G^T(W_i - S_w)G\|_F^2,$$

assuming G has orthonormal columns. Thus CPM maximizes

$$(2.3) \quad \sum_{i=1}^k \frac{N_i}{N} \|G^T(W_i - S_w)G\|_F^2,$$

which is the weighted variance of the k covariance matrices (after the projection G). Note that S_w is involved here, since $S_w = \sum_{i=1}^k \frac{N_i}{N} W_i$ is the weighted sum of $\{W_i\}_{i=1}^k$.

Following LDA, CPM considers the class discrimination (separation between different centroids), by maximizing $\text{trace}(G^T S_b G)$. Moreover, CPM attempts to also preserve class covariances. Specifically, CPM considers the information from both the class centroids and the class covariances simultaneously, via a tuning parameter α between 0 and 1. Mathematically, assuming S_t has been normalized, CPM computes the optimal G^* such that

$$(2.4) \quad G^* = \arg \max_{G: G^T G = I_d} h_\alpha(G),$$

where $0 \leq \alpha \leq 1$, and

$$\begin{aligned} h_\alpha(G) &= (1 - \alpha) \text{trace}(G^T S_b G) \\ &+ \alpha \sum_{i=1}^k \frac{N_i}{N} \|G^T(W_i - S_w)G\|_F^2. \end{aligned}$$

The optimization in (2.4) is nonlinear and difficult to solve. Instead, we maximize a lower bound, $f_\alpha(G)$, of $h_\alpha(G)$, where

$$\begin{aligned} f_\alpha(G) &= (1 - \alpha) \|G^T S_b^{\frac{1}{2}} G\|_F^2 \\ &+ \alpha \sum_{i=1}^k \frac{N_i}{N} \|G^T(W_i - S_w)G\|_F^2. \end{aligned}$$

It is easy to check that

$$\begin{aligned} \|G^T S_b^{\frac{1}{2}} G\|_F^2 &= \text{trace}(G^T S_b^{\frac{1}{2}} G G^T S_b^{\frac{1}{2}} G) \\ &\leq \text{trace}(G^T S_b G), \end{aligned}$$

since G has orthonormal columns. Thus,

$$f_\alpha(G) \leq h_\alpha(G).$$

Furthermore, we have the following result:

LEMMA 2.1. *Let S_b and G be defined above, then*

$$\begin{aligned} &\arg \max_{G: G^T G = I_d} \text{trace}(G^T S_b^{\frac{1}{2}} G G^T S_b^{\frac{1}{2}} G) \\ &= \arg \max_{G: G^T G = I_d} \text{trace}(G^T S_b G). \end{aligned}$$

Proof. Let

$$S_b = U \Sigma U^T$$

be the SVD of S_b , where $U \in \mathbb{R}^{n \times q}$ has orthonormal columns,

$$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_q), \quad \sigma_1 \geq \dots \geq \sigma_q > 0,$$

and $q = \text{rank}(S_b)$. Since $G^T G = I_d$, it can be shown [9] that

$$\text{trace}(G^T S_b G) \leq \sum_{i=1}^q \sigma_i,$$

and the equality holds when $G = U$. Next, we show that

$$U = \arg \max_{G: G^T G = I_d} \text{trace}(G^T S_b^{\frac{1}{2}} G G^T S_b^{\frac{1}{2}} G).$$

The proof follows from:

$$\text{trace}(G^T S_b^{\frac{1}{2}} G G^T S_b^{\frac{1}{2}} G) \leq \text{trace}(G^T S_b G) \leq \sum_{i=1}^q \sigma_i,$$

and

$$\text{trace}(U^T S_b^{\frac{1}{2}} U U^T S_b^{\frac{1}{2}} U) = \text{trace}(\Sigma^{\frac{1}{2}} \Sigma^{\frac{1}{2}}) = \sum_{i=1}^q \sigma_i.$$

Lemma 2.1 will be used in Proposition 2.1 below to show the relationship between CPM and LDA.

The optimization involved in CPM is thus approximated as finding G^* such that

$$(2.5) \quad G^* = \arg \max_{G: G^T G = I_d} f_\alpha(G).$$

2.2 The CPM algorithm Define $w_0 = 1 - \alpha$, $M_0 = S_b^{\frac{1}{2}}$, and $w_i = \alpha \frac{N_i}{N}$, $M_i = W_i - S_w$, for all $1 \leq i \leq k$. Then

$$\begin{aligned} f_\alpha(G) &= \sum_{i=0}^k w_i \|G^T M_i G\|_F^2 \\ (2.6) \quad &= \sum_{i=0}^k w_i \text{trace}((G^T M_i G)(G^T M_i G)). \end{aligned}$$

To our best knowledge, there is no closed form solution to the above maximization problem in (2.5) with $f_\alpha(G)$ as in (2.6) [5, 14]. Instead, we derive an iterative algorithm, which computes a sequence of matrices $G^{(j)}$, for $j \geq 1$. More specifically, $G^{(j+1)}$ is computed so that

$$G^{(j+1)} = \arg \max_{G: G^T G = I_d} \sum_{i=0}^k w_i \text{trace}(G^T M_i G^{(j)} (G^{(j)})^T M_i G).$$

It can be shown [14] that $G^{(j+1)}$ consists of the first d eigenvectors of

$$\sum_{i=0}^k w_i \left(M_i G^{(j)} (G^{(j)})^T M_i \right).$$

For simplicity, we choose $G^{(0)} = G_0 = (I_d, 0)^T$. However, empirical results show that CPM is insensitive to the initial choice as in [5, 14].

The main steps of the CPM algorithm include:

- (1) Normalize the data with $S_t = I_n$;
- (2) Initialization: $G^{(0)} \leftarrow G_0$, $j \leftarrow 0$, and $s_0 \leftarrow 0$;
- (3) Compute the first d eigenvectors $\{\phi_i\}_{i=1}^d$ of $\sum_{i=0}^k w_i (M_i G^{(j)} (G^{(j)})^T M_i)$;
- (4) $G^{(j+1)} \leftarrow [\phi_1, \dots, \phi_d]$;
- (5) $s_{j+1} \leftarrow \sum_{i=0}^k w_i \text{trace} \left((G^{(j+1)})^T M_i G^{(j)} (G^{(j)})^T M_i G^{(j+1)} \right)$;
- (6) Repeat (3)–(5) until $\frac{(s_{j+1} - s_j)}{s_{j+1}} \leq \eta$.

The convergence of CPM is determined by the threshold η . In our experiment, we choose $\eta = 10^{-6}$. The convergence of the CPM algorithm is established in the following theorem:

THEOREM 2.1. *Let $\{s_j\}_{j=0}^\infty$ be defined above. Then the sequence $\{s_j\}_{j=0}^\infty$ is nonincreasing and bounded from above. Thus the CPM algorithm converges.*

Proof. By the definition of s_j and s_{j+1} , we have

$$\begin{aligned} s_{j+1} &= \max_{G: G^T G = I_d} \sum_{i=0}^k w_i \text{trace} \left(G^T M_i G^{(j)} (G^{(j)})^T M_i G \right) \\ &\geq \sum_{i=0}^k w_i \text{trace} \left((G^{(j-1)})^T M_i G^{(j)} (G^{(j)})^T M_i G^{(j-1)} \right) \\ &= \sum_{i=0}^k w_i \text{trace} \left((G^{(j)})^T M_i G^{(j-1)} (G^{(j-1)})^T M_i G^{(j)} \right) \\ &= s_j. \end{aligned}$$

By the property of trace,

$$\begin{aligned} s_{j+1} &= \sum_{i=0}^k w_i \text{trace} \left((G^{(j+1)})^T M_i G^{(j)} (G^{(j)})^T M_i (G^{(j+1)}) \right) \\ &\leq \sum_{i=0}^k w_i \text{trace} (M_i M_i) = \sum_{i=0}^k w_i \|M_i\|_F^2. \end{aligned}$$

The CPM algorithm thus converges.

The relationship between CPM and LDA is described in Propositions 2.1 below.

PROPOSITION 2.1. *CPM is equivalent to LDA, if either of the following two conditions hold: (1) $\alpha = 0$; or (2) all classes have the same covariance matrix, i.e., $W_i = S_w$, for all i .*

Proof. Recall that in CPM, the optimal G is computed by maximizing $f_\alpha(G)$, defined as

$$f_\alpha(G) = (1-\alpha) \|G^T S_b^{\frac{1}{2}} G\|_F^2 + \alpha \sum_{i=1}^k \frac{N_i}{N} \|G^T (W_i - S_w) G\|_F^2,$$

subject to the constraint that $G^T G = I_d$.

When $\alpha = 0$, or $W_i = S_w$, for all i , the second term in $f_\alpha(G)$ vanishes. Thus,

$$\begin{aligned} \arg \max_{G: G^T G = I_d} f_\alpha(G) &= \arg \max_{G: G^T G = I_d} \|G^T S_b^{\frac{1}{2}} G\|_F^2 \\ &= \arg \max_{G: G^T G = I_d} \text{trace}(G^T S_b G), \end{aligned}$$

where the last equality follows from Lemma 2.1. This completes the proof of the proposition.

3 Relationship between CPM and SAVE and HDA

In this section, we study the relationship between CPM, SAVE, and HDA. More specifically, SAVE is shown to be closely related to a special case of CPM, and CPM and SAVE are shown to be approximations of HDA. The theoretical analysis provides the justification for the CPM algorithm and gives us insights into the nature of these different algorithms.

3.1 CPM versus SAVE Cook *et al.* [2, 3] proposed the Sliced Average Variance Estimator (SAVE) to overcome the limitations in LDA. Like CPM, each class in SAVE may have different covariances. That is, the covariances, W_i and W_j for the i -th and j -th classes ($i \neq j$) may be different. In their approach, the data is normalized so that the total covariance matrix is identity (as in CPM) and then the discriminant directions are chosen by maximizing

$$(3.7) \text{SAVE}(\alpha) = \alpha^T \left(\sum_{i=1}^k \left(\frac{N_i}{N} \right) (I_n - W_i) \right) \alpha,$$

over $\alpha \in \mathbb{R}^n$ of unit length, that is, $\|\alpha\| = 1$. The solutions are given by the top eigenvectors of

$$\sum_{i=1}^k \left(\frac{N_i}{N} \right) (I_n - W_i)^2.$$

Since $S_w + S_b = S_t = I_n$ (after normalization), we have

$$\sum_{i=1}^k \left(\frac{N_i}{N} \right) (I_n - W_i)^2 = \sum_{i=1}^k \left(\frac{N_i}{N} \right) (S_w - W_i + S_b)^2,$$

which equals to

$$\begin{aligned} & \sum_{i=1}^k \left(\frac{N_i}{N} \right) ((S_w - W_i)^2 + S_b^2 + (S_w - W_i)S_b \\ & + S_b(S_w - W_i)) = \sum_{i=1}^k \left(\frac{N_i}{N} \right) (S_w - W_i)^2 + S_b^2, \end{aligned} \quad (3.8)$$

where the last equality follows, since $\sum_{i=1}^k \left(\frac{N_i}{N} \right) = 1$ and $\sum_{i=1}^k \left(\frac{N_i}{N} \right) (S_w - W_i) = 0$.

Next, let us take a closer look at the CPM algorithm presented in Section 2. It was shown empirically in [5] that

$$\begin{aligned} & \arg \max_{G: G^T G = I_d} \sum_{i=1}^k w_i \|G^T M_i G\|_F^2 \\ & \approx \arg \max_{G: G^T G = I_d} \sum_{i=1}^k w_i \text{trace}(G^T M_i^2 G). \end{aligned}$$

Thus the solution to CPM can be approximated by maximizing

$$\begin{aligned} \tilde{f}_\alpha(G) &= (1 - \alpha) \text{trace}(G^T S_b G) \\ & + \alpha \sum_{i=1}^k \frac{N_i}{N} \text{trace}(G^T (W_i - S_w)^2 G), \end{aligned}$$

and is given by the first d eigenvectors of

$$(1 - \alpha)S_b + \alpha \sum_{i=1}^k \frac{N_i}{N} (W_i - S_w)^2,$$

which contains the same set of eigenvectors as

$$(3.9) \quad \sum_{i=1}^k \frac{N_i}{N} (W_i - S_w)^2 + S_b,$$

when $\alpha = \frac{1}{2}$.

Note that the matrices in (3.8) and (3.9) differ only in the second term. S_b^2 is used in SAVE, whereas S_b is used in CPM. Thus, SAVE is related to a special case of CPM when the parameter α is set to be 0.5. Our experimental results show that when $\alpha = 0.5$, CPM often has similar performance as SAVE. Recall that the parameter α controls the tradeoff between the separation of class centroids and the preservation of class covariances. CPM is more flexible in dealing with different

situations by varying the values of α . Experimental results in Section 4 show that CPM is competitive with SAVE and may significantly outperform SAVE for some cases.

3.2 CPM versus HDA LDA is optimal, when all classes have a common covariance matrix. However, the assumption is quite strict and may not applicable for some cases, as mentioned in Section 1. Kumar and Andreou [13] proposed Heteroscedastic Discriminant Analysis (HDA), which assumes each class is Gaussian, but with possibly different covariance matrices, under the assumption that both centroids and covariance matrices coincide in a subspace of the observation space. More specifically, HDA assumes that there exists an integer d , $d < n$, a full rank matrix G of dimension $n \times d$, and $\tilde{G} = [G, F] \in \mathbb{R}^{n \times n}$, such that

$$(3.10) \quad \tilde{G}^T c_i = \begin{pmatrix} G^T c_i \\ F^T \tilde{c} \end{pmatrix}$$

and

$$(3.11) \quad \tilde{G}^T W_i \tilde{G} = \begin{pmatrix} G^T W_i G & 0 \\ 0 & F^T \tilde{W} F \end{pmatrix},$$

for some \tilde{c} and \tilde{W} , which are common for all i .

This implies that HDA assumes that the differences between classes lie solely in a subspace of d dimensions (projection by G), whereas in the complementary subspace of $n - d$ dimensions (projection by F) they are identical, i.e. useless for discrimination.

The probability density of $a_i \in \mathbb{R}^n$ under the above model is given as

$$\begin{aligned} P(a_i) &= \frac{\det(\tilde{G})}{\sqrt{(2\pi)^n |\tilde{G}^T W_{g(i)} \tilde{G}|}} \exp \left(-\frac{1}{2} \left(\tilde{G}^T a_i - \tilde{G}^T c_i \right)^T \right. \\ & \quad \left. \left(\tilde{G}^T W_{g(i)} \tilde{G} \right)^{-1} \left(\tilde{G}^T a_i - \tilde{G}^T c_i \right) \right), \end{aligned}$$

where a_i belongs to the class $g(i)$.

The log-likelihood of the data under the linear transformation \tilde{G} and under the constrained Gaussian model above is

$$\begin{aligned} \log L &= \sum_{i=1}^N \log P(a_i) \\ &= -\frac{1}{2} \sum_{i=1}^N \left\{ \left(\tilde{G}^T a_i - \tilde{G}^T c_i \right)^T \left(\tilde{G}^T W_{g(i)} \tilde{G} \right)^{-1} \right. \\ & \quad \left. \left(\tilde{G}^T a_i - \tilde{G}^T c_i \right) + \log \left((2\pi)^n |\tilde{G}^T W_{g(i)} \tilde{G}| \right) \right\} \\ (3.12) \quad & + N \log \det(\tilde{G}). \end{aligned}$$

The above likelihood function can then be maximized with respect to its parameters. Since there is no closed-form solution for maximizing the likelihood with respect to its parameters, the maximization has to be performed numerically. More details on this can be found in [13], where quadratic programming is performed for the required optimization. As pointed out in [13], even though the quadratic-optimization techniques are used, the likelihood surface is not strictly quadratic, and the optimization problem occasionally fails. We often observe this problem in our experiments.

In the rest of this section, we show that the CPM algorithm proposed in this paper is an approximation of HDA.

The first assumption in (3.10) requires that

$$\tilde{G}^T(c_i - c) = \begin{pmatrix} G^T(c_i - c) \\ 0 \end{pmatrix},$$

i.e., most of the information on $c_i - c$ is preserved in the subspace spanned by G . It follows that G is computed so that

$$\sum_{i=1}^k \frac{N_i}{N} \|G^T(c_i - c)\|^2 = \text{trace}(G^T S_b G)$$

is large.

The second assumption in (3.11) requires that

$$\tilde{G}^T(W_i - S_w)\tilde{G} = \begin{pmatrix} G^T(W_i - S_w)G & 0 \\ 0 & 0 \end{pmatrix}.$$

Similarly, this implies that

$$\sum_{i=1}^k \frac{N_i}{N} \|G^T(W_i - S_w)G\|_F^2$$

is large.

These two assumptions can simultaneously be approximated by maximizing $h_\alpha(G)$ in (2.4). Thus, CPM can be considered as an approximation of HDA. Our experimental results in the next section show that CPM is comparable to HDA in classification, while CPM is much more efficient and robust than HDA.

3.3 SAVE versus CPM SAVE can also be considered as an approximation of HDA, based on the following result.

LEMMA 3.1. *Let S_b be defined above and let $G \in \mathbb{R}^{n \times d}$ with $d < n$. Then*

$$\begin{aligned} & \max_{G^T G = I_d} \{ \text{trace}(G^T S_b G) \} \\ & = \max_{G^T G = I_d} \{ \text{trace}(G^T S_b^2 G) \}. \end{aligned}$$

Proof. Let

$$S_b = U \Sigma U^T$$

be the SVD of S_b , where $U \in \mathbb{R}^{n \times q}$ has orthonormal columns,

$$\Sigma = \text{diag}(\sigma_1, \dots, \sigma_q), \quad \sigma_1 \geq \dots \geq \sigma_q > 0,$$

and $q = \text{rank}(S_b)$. From Lemma 2.1,

$$\text{trace}(G^T S_b G) \leq \sum_{i=1}^q \sigma_i,$$

and the equality holds when $G = U$. Similarly,

$$\text{trace}(G^T S_b^2 G) = \text{trace}(G^T U \Sigma^2 U^T G) \leq \sum_{i=1}^q \sigma_i^2,$$

and the maximum value is achieved when $G = U$. This completes the proof of the lemma.

Lemma 3.1 shows that the two assumptions of HDA are approximated by maximizing the criterion of SAVE in (3.8). While CPM applies the weight α to balance the tradeoff between these two assumptions, SAVE puts an equal weight to both terms.

4 Experiments

In this section, we evaluate CPM on both synthetic and real-world datasets. The 1-Nearest-Neighbor algorithm is applied for classification. For simplicity, $\alpha = 0.2$ is used for CPM in all experiments. However, the best value of α may be estimated through cross-validation.

The first experiment is on a synthetic dataset, where the class centroids of three classes do not coincide, but they have different covariance matrices. For the first two coordinates, class 1 is simulated from a standard multivariate Gaussian with mean $(0, 0)$ and covariance $C = \text{diag}(2, 2)$. Classes 2 and 3 are mixtures of two shifted standard Gaussians, with mean $(-11, 0)$, $(5, 0)$ and $(-5, 0)$, $(-11, 0)$ respectively. Each Gaussian component also has covariance C . From Fig. 2, HDA, SAVE, and CPM separate the three classes better than LDA, which shows the effect of incorporating the class covariance information. LDA does not consider the class covariance information and fails to find the best projection, when class covariances vary.

Our next experiment is on the Pendigits dataset from the UCI machine learning repository.¹ It contains the (x, y) coordinates of hand-written digits. Each digit is represented as a vector in 16-dimensional space. The dataset is divided into a training set (7494 digits) and a

¹<ftp://ftp.ics.uci.edu/pub/machine-learning-databases/>

test set (3498 digits). For the purpose of visualization as in [16], we select the 0's, 6's and 9's. We apply LDA, SAVE, HDA, and CPM to this 3-class subproblem (which contains 2219 training digits and 1035 test digits) and extract 2 leading discriminant directions. We then project the test set onto these 2 dimensions. The results are shown in Fig. 3. It is clear that LDA, HDA, and CPM separate the three classes well. SAVE, on the other hand, picks up directions where one class has a much larger variance than the other two classes, as mentioned in [16]. This further confirms our claim in Section 3 that CPM is more flexible in dealing with different situations by choosing different α , than SAVE, where α is fixed to be 0.5. Note that we also run CPM with $\alpha = 0.5$. The result is similar to SAVE and is thus omitted.

Next, we evaluate CPM in terms of classification on two datasets: Pendigits and Ionosphere, both from the UCI machine learning repository, using different number, d , of reduced dimensions. For Pendigits, we use a subset consisting of 0's, 5's, 6's, and 9's. Ionosphere is a binary dataset consisting of 351 instances of radar collected data, with 34 dimensions (200 instances for training and 151 instances for test). The result is summarized in Fig. 4, where the x -axis denotes the reduced dimension d for HDA, SAVE, and CPM, and the y -axis denotes the classification accuracy. The result on LDA with the fixed reduced dimension, $k - 1$, is also presented. We ran HDA on the Ionosphere dataset but the algorithm did not complete. This may be due to the complex optimization problem involved in HDA, as mentioned in Section 3. We observe from Fig. 4 that the best accuracy for HDA, SAVE, and CPM often occurs when $d > k - 1$. In practice, the best dimension d can be estimated through cross-validation. Also note that the accuracy curves of HDA and CPM follow similar trend on the Pendigits dataset (see Section 3 for theoretical analysis), while they are quite different from SAVE. Overall, CPM is very competitive with the other three algorithms.

Finally we compare CPM with SAVE and LDA in terms of classification on four datasets from UCI. The results are summarized in Table 1. The reduced dimension used in CPM is set to be $k - 1$, the reduced dimension used in LDA. Note that based on our previous experiments, the performance of CPM may be improved by using larger number of dimensions. HDA is not included in the comparison, since it is not able to complete for many of the datasets. We can observe that CPM outperforms SAVE for all cases, while CPM is very competitive with LDA. We can also observe that LDA performs reasonably well for all datasets. This implies that LDA is fairly robust in practice, even though the

assumption involved in LDA may be violated.

5 Conclusions

A new algorithm, namely CPM is proposed for dimension reduction. It aims to maximize the class discrimination and preserve the class covariance simultaneously. The optimization problem involved in CPM is equivalent to low rank approximations of a collection of matrices, which can be solved iteratively. We have applied CPM to various datasets and our empirical results show that: (1) CPM is capable of recovering the features for classification, even when all class centroids coincide or covariance matrices vary, overcoming limitations of LDA; and (2) CPM is competitive with LDA, SAVE, and HDA, in terms of classification.

Our theoretical analysis reveals the close relationship between CPM and LDA, SAVE, and HDA. LDA is shown to be a special case of CPM, which generalizes LDA by utilizing the class covariance information. CPM and SAVE are shown to be approximations of HDA, while they are much more efficient and robust than HDA. SAVE is closely related to a special case of CPM, when the parameter α , which controls the tradeoff between the separation of class centroids and the preservation of class covariances, is set to be 0.5. CPM is thus more flexible in dealing with different situations. These theoretical results further justify the criterion used in CPM and give us new insights into these algorithms.

Some directions for future work include: (1) extension of CPM to deal with high-dimensional data; and (2) extension of CPM to deal with nonlinear data using kernels.

Acknowledgements

Research of JY was sponsored, in part, by the Evolutionary Functional Genomics Center of the Biodesign Institute at the Arizona State University. Support Fellowships from Guidant Corporation and from the Department of Computer Science & Engineering, at the University of Minnesota, is gratefully acknowledged.

References

- [1] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
- [2] R. D. Cook and S. Weisberg. Discussion of Li (1991). *Journal of the American Statistical Association*, 86: 1328–332, 1991.
- [3] R. D. Cook and X. Yin. Dimension reduction and visualization in discriminant analysis (with discussion).

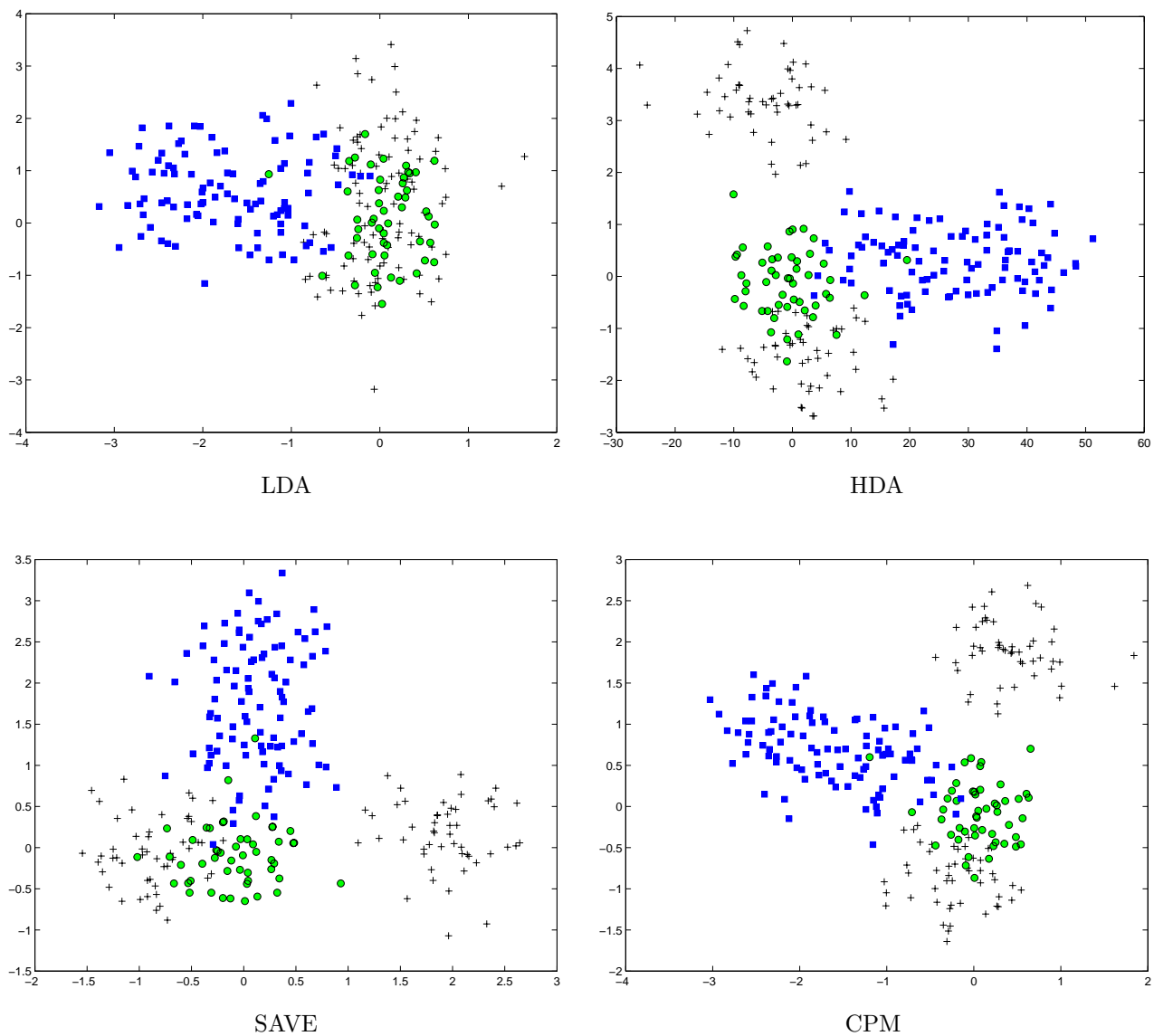
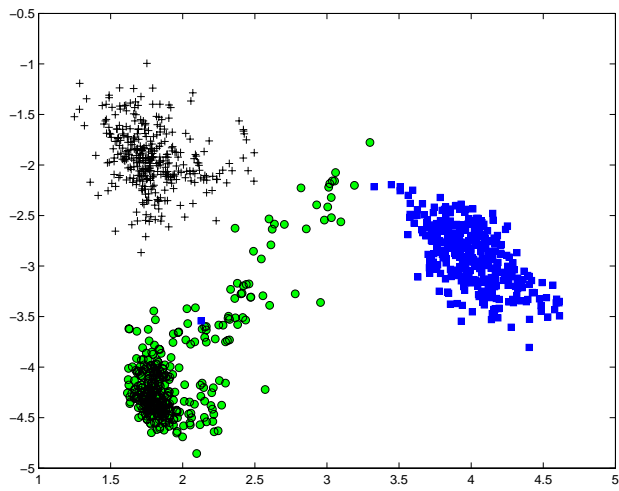
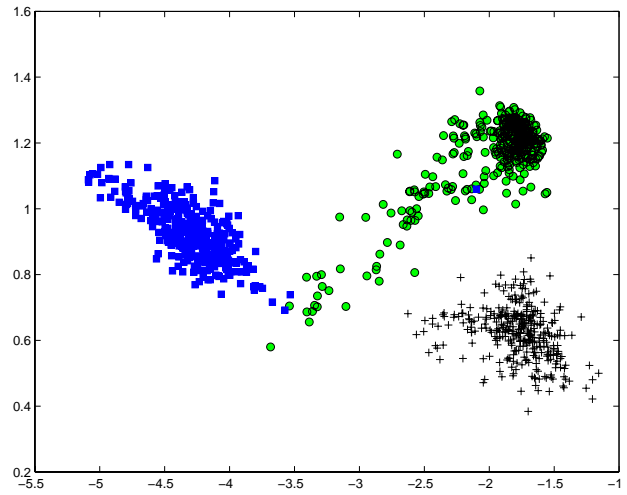


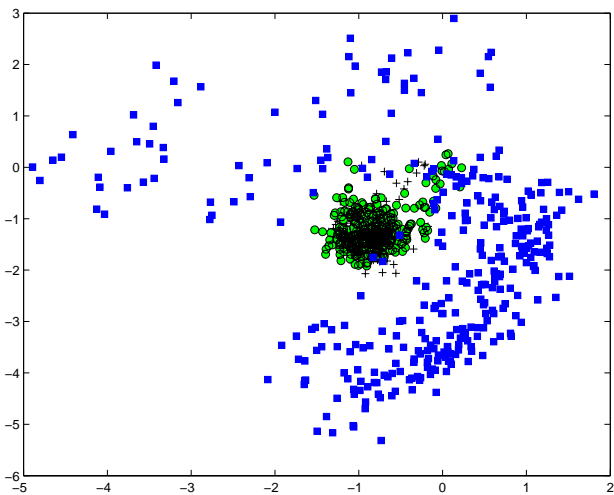
Figure 2: Projections by LDA, HDA, SAVE and CPM using the synthetic dataset, where the class centroids of three classes do not coincide, but they have different covariance matrices. HDA, SAVE and CPM consider the class covariance information and separate the three classes better than LDA.



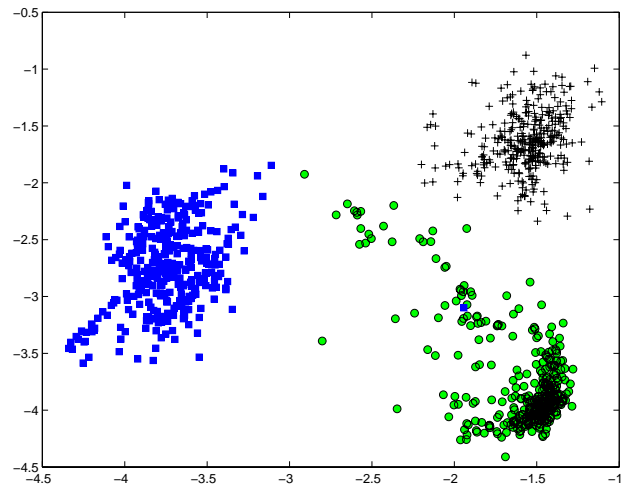
LDA



HDA



SAVE



CPM

Figure 3: Projections by LDA, HDA, SAVE and CPM using the Pendigits dataset. LDA, HDA, and CPM separate the three classes better than SAVE.

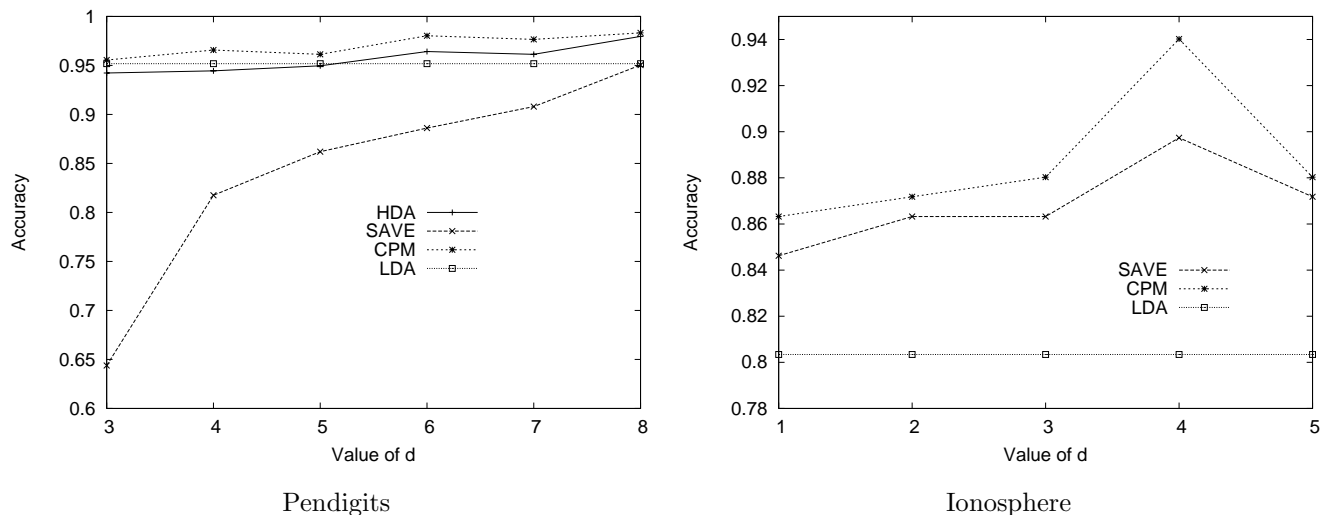


Figure 4: Comparison of classification accuracy of LDA, HDA, SAVE and CPM using the Pendigit and Ionosphere datasets.

dataset	CPM	SAVE	LDA
Pima	69.55% (0.029)	59.68% (0.030)	68.87% (0.028)
Vote	73.13% (0.048)	51.59% (0.050)	72.72% (0.048)
Waveform	80.13% (0.021)	79.18% (0.022)	76.16% (0.026)
Ionosphere	86.32% (0.016)	84.62% (0.018)	80.34% (0.022)

Table 1: Comparison on classification accuracy and standard deviation (in parenthesis).

- Australian and New Zealand Journal of Statistics*, 43 (2):147–199, 2001.
- [4] R.O. Duda, P.E. Hart, and D. Stork. *Pattern Classification*. Wiley, 2000.
- [5] C. Ding and J. Ye. 2-Dimensional singular value decomposition for 2D maps and images. In *SIAM Data Mining Conference Proceedings*, 2005.
- [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] K. Fukunaga. *Introduction to Statistical Pattern Classification*. 1990.
- [8] T. Hastie, R. Tibshirani, and J.H. Friedman. *The elements of statistical learning : data mining, inference, and prediction*. Springer, 2001.
- [9] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Third edition, 1996.
- [10] T. Hastie and R. Tibshirani. Discriminant analysis by Gaussian mixtures. *Journal of the Royal Statistical Society series B*, 58:158–176, 1996.
- [11] P. Howland, M. Jeon, and H. Park. Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 25(1): 165–179, 2003.
- [12] I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.
- [13] N. Kumar and A.G. Andreou. A generalization of linear discriminant analysis in maximum likelihood framework. In *Proceedings of Joint Meeting of ASA*, 1996.
- [14] J. Ye. Generalized low rank approximations of matrices. In *ICML Conference Proceedings*, pages 887–894, 2004.
- [15] J. Ye. Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems. *Journal of Machine Learning Research*, 6: 483–502, 2005.
- [16] M. Zhu and T. Hastie. Feature extraction for non-parametric discriminant analysis. *Journal of Computational and Graphical Statistics*, 12(1):101–120, 2003.