Bayes Optimality in Linear Discriminant Analysis

Onur C. Hamsici and Aleix M. Martinez
Department of Electrical and Computer Engineering
The Ohio State University, Columbus, OH 43210

Abstract

We present an algorithm which provides the one-dimensional subspace where the Bayes error is minimized for the C class problem with homoscedastic Gaussian distributions. Our main result shows that the set of possible one-dimensional spaces $\mathbf{v}$, for which the order of the projected class means is identical, defines a convex region with associated convex Bayes error function $g(\mathbf{v})$. This allows for the minimization of the error function using standard convex optimization algorithms. Our algorithm is then extended to the minimization of the Bayes error in the more general case of heteroscedastic distributions. This is done by means of an appropriate kernel mapping function. This result is further extended to obtain the d-dimensional solution for any given d, by iteratively applying our algorithm to the null space of the $(d-1)$-dimensional solution. We also show how this result can be used to improve upon the outcomes provided by existing algorithms, and derive a low-computational cost, linear approximation. Extensive experimental validations are provided to demonstrate the use of these algorithms in classification, data analysis and visualization.

Index terms: Linear discriminant analysis, feature extraction, Bayes optimal, convex optimization, pattern recognition, data mining, data visualization.

1 Introduction

A major goal in pattern recognition is to define an algorithm that can find those few dimensions that best discriminate a set of C classes, and where it is generally assumed that each class is represented as a Gaussian distribution or mixture of these. In his ground-breaking work, Fisher [4] defined an optimal algorithm to find that 1-dimensional subspace that minimizes the Bayes error in the 2-class problem under the assumption that all Gaussians have a common covariance matrix (i.e., homoscedastic distributions). In subsequent work, Rao [18] proposed an extension of the algorithm to find that $(C-1)$-dimensional subspace which minimizes the Bayes error for the C class homoscedastic problem. This solution is known as (Fisher) Linear Discriminant Analysis (LDA or FDA, for short).

Unfortunately, LDA is not guaranteed to find the optimal subspace of dimensionality d strictly smaller than $C-1$. An example is given in Fig. 1. In (a) we show six 3-dimensional homoscedastic Gaussian distributions with distinct mean. In this particular case, LDA can find that 3-dimensional subspace where the data is best separated; but this proves useless, because this result is exactly the same as the original space. In many applications, one desires to find the one-dimensional subspace where the classes are best separated (i.e., the Bayes error is minimized). This solution is shown in (b). For comparison, the most discriminant basis found by LDA is plotted in (c). As one can see, LDA's solution is far from optimal.

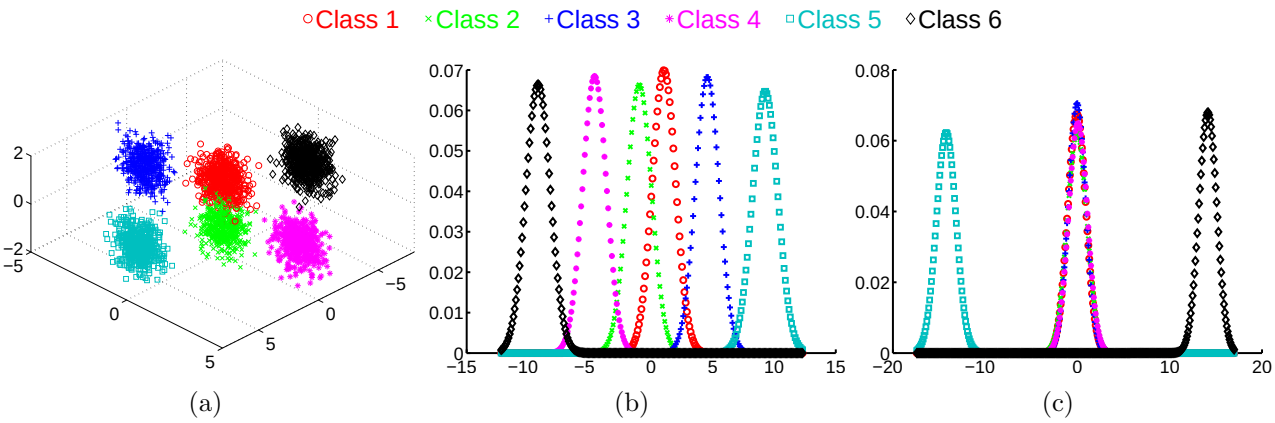


Figure 1: *(a) Shown here are six homoscedastic Gaussian distributions in* $\mathbb{R}^3$*. (b) Shows the Bayes optimal 1-dimensional result. The result obtained with the classical Fisher-Rao LDA algorithm is illustrated in (c), for comparison.*

The problem mentioned in the preceding paragraph, has restricted the uses of LDA enormously, especially so in problems in biology, medicine, psychology, and anthropology, where the goal is to find that one or two dimensional space which best classifies a set of classes [1, 3, 19, 21, 22, 14]. As an example, imagine the following problem in medicine. It is our task to find that combination of factors which are responsible for the C different types of an illness observed in a set of n patients. Here, we want to find that one-dimensional space which identifies the combination of causes producing the n observations. A $C-1$ set of potential combinations would result of little practical use, because it would not allow us to determine the real causes but would only provide a set of possible cause-effect hypotheses. Similarly, in genomics, one would like to find that single combination of gene mutations that cause a set of C subclasses of a disease (e.g., in leukemia where distinct subgroups may require different treatments), rather than a space of $C-1$ dimensions where the underlying causes lay. And, in psychology and anthropology, we usually want to visualize a group of n samples belonging to C classes in a 2-dimensional plot from which conclusions can be drawn.

In engineering, discriminant analysis has a large number of potential applications too. In computer vision and speech analysis, for example, one desires to find the smallest space where the class distributions can be efficiently separated. This is needed because learning algorithms are generally much more efficient when working on low-dimensional representations. The smaller our representation is, the better and faster the algorithm will generally work [12]. This is similar to problems in the physical sciences where the goal is to find those features that best explain a set of observations or those which validate or invalidate an existing model. Furthermore, the need for such algorithms is expected to increase with the new particle accelerator, the large Hardon collider, developed at CERN, which is expected to open its doors in 2007.

In this paper, we present an algorithm to find the one-dimensional subspace where a set of C homoscedastic Gaussian distributions is optimally separated (meaning, where the Bayes error is minimized) for any value of $C > 1$. This is in contrast to other methods defined to date (including Fisher-Rao's LDA) which, in general, can only find the (Bayes) optimal solution when $C = 2$ (see Section 2 for a formal definition of the problem). We then extend our basic formulation to the heteroscedastic case by applying an intrinsic mapping function in the form of a kernel. This permits the use of our algorithm in a large number of potential real applications as we will demonstrate in the experimental results section.

This search for the Bayes optimal solution is made possible thanks to the realization that the Bayes error function defined for each possible ordered set of means on $\mathbf{v}$ is convex (here, $\mathbf{v} \in \mathbb{R}^p$

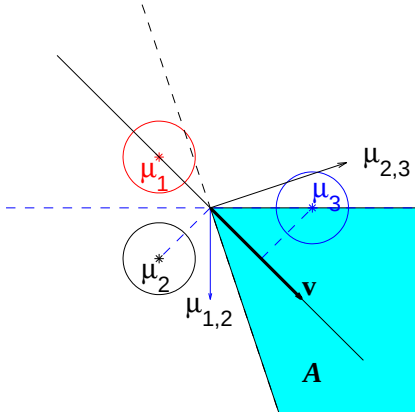


Figure 2: *This figure shows a basis* $\mathbf{v}$ *where the sequence of projected, whitened class means is* $\eta_1 \leq \eta_2 \leq \eta_3$. *Now, note that the color filled region defines the area with all those bases* $\mathbf{v}$ *producing the same sequence. More accurately, this region defines a convex polyhedra given by the following constraint equations* $\mathbf{v}^T\mu_{1,2} \geq 0$ *and* $\mathbf{v}^T\mu_{2,3} \geq 0$, *where* $\mu_{i,j} = \mu_j - \mu_i$ *(see Theorem 1). Also note that the boundaries of this region* $\mathcal{A}$ *are given by the hyperplanes orthogonal to* $\mu_{i,j}$.

represents the 1-dimensional subspace). To realize this, we first note that the class means projected onto the subspace defined by $\mathbf{v}$, $\eta_i = \mathbf{v}^T\mu_i$ (where μ_i is the mean of class i), are ordered in the same manner for a range of values of $\mathbf{v}$. This is illustrated in Fig. 2, where the color filled area represents the set of all 1-dimensional subspaces $\mathbf{v}$ where the order of the projected class means is $\eta_1 \leq \eta_2 \leq \eta_3$. It is important to note that this region is convex. Further, as we show in Section 3, the Bayes error function within this region is also convex. This means that, for every ordered sequence of projected class means, we can find the 1-dimensional subspace where the Bayes error is minimized by using standard convex optimization algorithms. In Section 4, we extend this to find the d-dimensional subspace. This is achieved by iteratively applying the algorithm to the null space of the current solution. In that section, we also derive a set of approximations that can be readily employed to find a close-to-optimal solution very fast (i.e., with associated low computational cost). Such extensions are necessary to make general approaches applicable to the large number of problems outlined above. Experimental results are in Section 5. Conclusions are in Section 6.

2 The Non-optimality of LDA

We start with the classical definition of a multivariate Gaussian (Normal) distribution $N_i(\mu_i, \Sigma_i)$ with mean μ_i and covariance matrix Σ_i, $f_i(\mathbf{x}|\Sigma_i, \mu_i) = \frac{1}{\sqrt{(2\pi)^d|\Sigma_i|}} \exp(-(\mathbf{x}-\mu_i)^T\Sigma_i^{-1}(\mathbf{x}-\mu_i)/2)$. A group of C Gaussian distributed classes, $N_i(\mu_i, \Sigma_i)$ with $i = \{1, \ldots, C\}$, is called homoscedastic, if their class covariances are all the same, $\Sigma_i = \Sigma_j$, $\forall i, j \in \{1, \ldots, C\}$. Given these Gaussian distributions and their class prior probabilities $p_i(\mathbf{x})$, the classification of a test sample $\mathbf{t}$ can be obtained by comparing the log-likelihoods of $f_i(\mathbf{t}|\mu_i, \Sigma_i)p_i(\mathbf{t})$ for all i; that is, $rec = \arg\min_{1\leq i\leq C} d_i(\mathbf{t})$, where

$$d_i(\mathbf{t}) = (\mathbf{t}-\mu_i)^T\Sigma_i^{-1}(\mathbf{t}-\mu_i) + ln|\Sigma_i| - 2\ln p_i(\mathbf{t}) \tag{1}$$

is known as the discriminant score of class i, and rec is the class label of $\mathbf{t}$ identified by the algorithm, $rec \in [1, C]$.

In general this discriminant score leads to quadratic classification boundaries between classes. However, for the case where the class distributions have the same covariance matrix (i.e. $\Sigma = \Sigma_i$,

$\forall i \in \{1, \ldots, C\}$), the quadratic parts in (1) cancel out leading to optimal *linear* classifiers also known as linear discriminant bases – hence the name Linear Discriminant Analysis (LDA). Furthermore, since the classifiers are linear, the discriminant scores can be calculated without any loss in a $(C-1)$-dimensional subspace spanned by the mean vectors in the whitened space of the (common) covariance matrix Σ. This means that, in the homoscedastic case, Fisher's LDA provides the Bayes optimal classification in this subspace of $C-1$ dimensions. Recall that the means in the whitened space are given by $\hat{\mu}_i = \Lambda^{-1/2}\mathbf{V}^T\mu_i$, where $\mathbf{V}$ and Λ are the eigenvector and eigenvalue matrices of Σ.

Unfortunately, there is no known way to find the Bayes optimal subspace of d dimensions when $d < C-1$. Rao [18] defined an intuitive algorithm that provides an approximation to this optimal solution. In his approach, we first define a covariance matrix of the class means on the whitened space, usually referred to as between-class scatter matrix $\hat{\mathbf{S}}_B = \sum_{i=1}^{C}(\hat{\mu}_i - \hat{\mu})(\hat{\mu}_i - \hat{\mu})^T$, where $\hat{\mu}$ is the average over all $\hat{\mu}_i$, and $\hat{\mu}_i$ are the class means in the whitened space. Then, the eigenvector associated to the largest eigenvalue of $\hat{\mathbf{S}}_B$ is assumed to define the 1-dimensional subspace where the data distributions are best separated. Since the first eigenvector is the one that minimizes the mean-square error (i.e., least-squares solution), this is only equal to the Bayes optimal basis when the farthest mean defines the direction of best separability [13].

The non-optimality of LDA for $C > 2$ and subspaces of dimensionality lower than $(C-1)$ has been pointed out by several researchers. Geisser [6] showed that the Bayes error for C homoscedastic classes with equal priors projected onto a single dimension can be calculated as

$$2\,C^{-1}\sum_{i=1}^{C-1}\Phi\left(\frac{\eta_{(i)} - \eta_{(i+1)}}{2}\right), \tag{2}$$

where $\Phi(.)$ is the cumulative distribution function (cdf) of a standard Normal distribution with unit variance, $\eta_{(i)}$ are the ordered elements η_i such that $\eta_{(1)} \leq \eta_{(2)} \leq \cdots \leq \eta_{(C)}$, and

$$\eta_i = \frac{\mathbf{v}^T\mu_i}{\sqrt{\mathbf{v}^T\Sigma\mathbf{v}}}$$

are the mean feature vectors μ_i whitened and projected onto the one-dimensional subspace defined by the unit vector $\mathbf{v}$. Using the same notation, one can show that LDA maximizes $\sum_{i=1}^{C}(\eta_i - \eta)^2$, where η is the average of the projected, whitened means η_i. Therefore, the LDA result is clearly not the same as the Bayes optimal solution minimizing (2).

Geisser, however, does not propose a solution to this problem. Schervish [20] shows that, if the original space is $\mathbb{R}^2$, the Bayes risk of the 3-class problem can be defined as a convex function which depends on the angle of the 1-dimensional solution. Unfortunately, this method has not been successfully extended beyond $\mathbb{R}^2$ or $C > 3$, and a general solution has remained elusive.

Due to the difficulty associated with finding the optimal solution in the case where $C > 2$, researchers have turned to approximations. One such approximation is Nonparametric DA (NDA) [5], where the between-class scatter matrix is modified to adapt to data drawn from general non-parametric densities. This generalizes the use of LDA. In [9], the authors take advantage of the knowledge that Fisher's solution is biased toward those class means that are farthest apart to define the approximate Pairwise Accuracy Criteria (aPAC). Here, the authors define a new weighted between-class scatter matrix where each vector $\hat{\mu}_i - \hat{\mu}_j$ contributing to $\hat{\mathbf{S}}_B$ is assigned a weight proportional to the Bayes accuracy between classes i and j in that dimension. This means that the first dimensions are usually not as biased as in LDA. In a similar approach, Lotlikar and Kothari [10] present a fractional-step dimensionality reduction algorithm (herein, referred to as Fractional LDA or FLDA for short) which attempts to reduce the bias of the between-class scatter by assigning

to it weights that are inversely proportional to the distance between each class mean pair. This algorithm can be implemented iteratively, where one of the dimensions of $\hat{\mathbf{S}}_B$ is eliminated at each iteration, resulting finally in any d-dimensional solution one may desire to have. When doing so, care is taken to prevent pair of class means to overlap on the space of reduced dimensionality. This solution is optimal when $d = C - 1$ and an approximation elsewhere. FLDA may still fail when the samples to dimensionality ratio is small. To address this problem, DFLDA (Direct Fractional LDA) was proposed in [11]. In this algorithm, the data is first projected onto the subspace of the between-class scatter matrix to simplify the complexity of the other steps in the algorithm.

In what follows, we turn to the Bayes optimal solution in the 1-dimensional subspace and its extensions to $d > 1$ dimensions. Our goal is to find that set of d basis vectors where the Bayes error in the first basis is minimized, the Bayes error in the second basis is minimized with the added constraint that this vector is orthogonal to the previous solution, and so on.

3 The Bayes Optimal One-dimensional Subspace

Since our main result is applicable for homoscedastic Gaussian distributions, throughout this section we assume that we are working in the whitened space, where all covariance matrices are equal to the identity matrix, $\widehat{\Sigma} = \mathbf{I}$. For simplicity of notation, we further assume that all class distributions have the same prior. Our first main result defines under which conditions the Bayes error function is convex. This will then be used to derive our first algorithm.

Theorem 1. *Define a constrained region $\mathcal{A}$ where all vectors $\mathbf{v}$ sampled from it generate the same ordered sequence $\eta_{(i)}$ of the whitened, projected mean locations $\eta_i = \mathbf{v}^T\hat{\mu}_i$ of C homoscedastic Gaussian distributions. Moreover, let $g(\mathbf{v})$ be the Bayes error function of the C homoscedastic Gaussian distributions in $\mathcal{A}$. Then, the region $\mathcal{A}$ is a convex polyhedron, and the Bayes error function $g(\mathbf{v})$ for all $\mathbf{v} \in \mathcal{A}$ is also convex.*

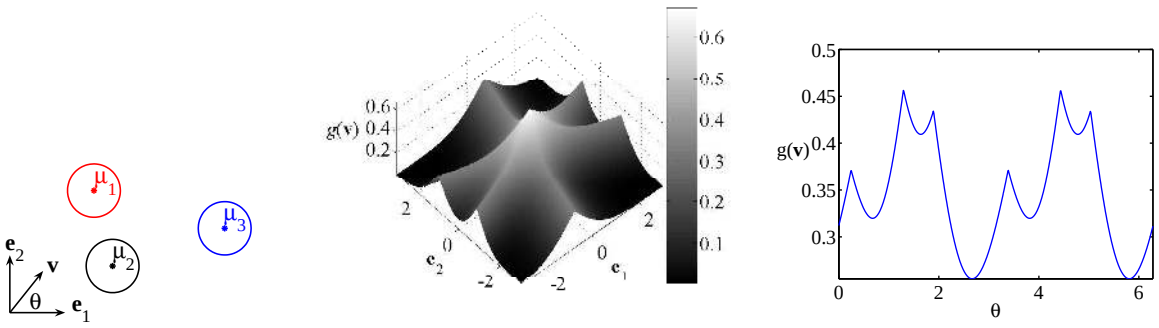
Proof. Without loss of generality we can assume the ordered sequence $\eta_{(i)}$ is given by $\eta_{(i)} = \eta_i$ (the same applies to any other ordering). Under this assumption, the ordered sequence is $\eta_1 \leq \eta_2 \leq \cdots \leq \eta_C$. The inequality $\eta_1 = \mathbf{v}^T\hat{\mu}_1 \leq \eta_2 = \mathbf{v}^T\hat{\mu}_2$ defines a positive half-space (i.e., the positive half part of the space with regard to the normal vector of the dividing hyperplane) given by $\mathbf{v}^T(\hat{\mu}_2 - \hat{\mu}_1) = \mathbf{v}^T\hat{\mu}_{1,2} \geq 0$. More generally, all the inequalities in this ordered sequence define the intersection of $(C-1)$ half-spaces, i.e.,

$$\mathbf{v}^T\hat{\mu}_{1,2} \geq 0, \ldots, \mathbf{v}^T\hat{\mu}_{C-1,C} \geq 0. \tag{3}$$

The intersection of the half-spaces given by these $C-1$ inequalities is our region $\mathcal{A}$. Note that the intersection of these half-spaces will always include the origin.[1] More generally, by definition, it corresponds to a convex polyhedron (see also Fig. 2).

Next, note that $\eta_{(i+1)} - \eta_{(i)} = \eta_{i+1} - \eta_i = |\mathbf{v}^T\hat{\mu}_{i,i+1}| = \mathbf{v}^T\hat{\mu}_{i,i+1}$, where the last equality holds because $\mathbf{v}$ is constrained to be in a convex region defined by $\mathbf{v}^T\hat{\mu}_{i,i+1} \geq 0$. (This argument also extends to all other ordered sequences not considered here, because the positivity of $\mathbf{v}^T\hat{\mu}_{j,k}$ will be guaranteed by the constraints defining the convex region for that sequence; where $\hat{\mu}_{j,k}$ is given by the order of projected means in $\mathbf{v}$, that is, if $\hat{\mu}_j$ corresponds to $\eta_{(i)}$ then $\hat{\mu}_k$ corresponds to $\eta_{(i+1)}$.)

[1] In some cases, the origin may actually be the only component of the resulting convex region $\mathcal{A}$. This may happen when the data is in a space of smaller dimensionality than $C-1$. We note that this then means that not all the orderings of the projected means are possible.



(a) (b) (c)

Figure 3: *(a) Shown here is a simple three-class problem in the two-dimensional whitened space.* $(\mathbf{e}_1, \mathbf{e}_2)$ *are the bases of this whitened space. The goal is to find that* θ *providing that* $\mathbf{v}$ *where the Bayes error is minimized. (b) Plot of the values of* $g(\mathbf{v})$ *for each of its possible values in* $\mathbf{e}_1$ *and* $\mathbf{e}_2$*. For visualization purposes we restricted the plot for values within* $[-3, 3]$*. (c) When restricting the search to unit vectors, the problem reduces to finding the* θ *that minimizes* $g(\mathbf{v})$*. Each of the minima seen in this plot can be determined by solving the convex optimization problem described in the text.*

It then follows that

$$g(\mathbf{v}) = \frac{2}{C} \sum_{i=1}^{C-1} \Phi\left(\frac{-\mathbf{v}^T \hat{\mu}_{i,i+1}}{2}\right).$$

The gradient of this function with respect to $\mathbf{v}$ is

$$\nabla g(\mathbf{v}) = -\frac{1}{C} \sum_{i=1}^{C-1} \frac{1}{\sqrt{2\pi}} \exp(-z^2/2) \hat{\mu}_{i,i+1},$$

where $z = -\mathbf{v}^T \hat{\mu}_{i,i+1}/2$. And the Hessian is

$$\nabla^2 g(\mathbf{v}) = \frac{1}{4C\sqrt{2\pi}} \sum_{i=1}^{C-1} \exp(-z^2/2)(\mathbf{v}^T \hat{\mu}_{i,i+1})\ \hat{\mu}_{i,i+1}\ \hat{\mu}_{i,i+1}^T.$$

Since the coefficients of $\hat{\mu}_{i,i+1}\ \hat{\mu}_{i,i+1}^T$ are positive semi-definite in $\mathcal{A}$ (i.e., $\exp(-z^2/2) \geq 0$ and $(\mathbf{v}^T \hat{\mu}_{i,i+1}) \geq 0$), the Hessian of the Bayes error function is positive semi-definite too. This means that the Bayes error function $g(\mathbf{v})$ is convex in $\mathcal{A}$. □

Our main result given above provides a mechanism to find the solution for any given ordered sequence of projected means $\eta_{(1)} \leq \eta_{(2)} \leq \cdots \leq \eta_{(C)}$. This reduces to solving a convex optimization problem. The next important problem is to determine *which of all possible sequences that one can have in a 1-dimensional space provides the optimal solution* where the Bayes error between Gaussians (classes) is minimized. This is illustrated in Fig. 3.

From Theorem 1 we note that the number of possible solutions is the number of possible sequences, which is $C!$. Closer analysis, reveals that the mirrored sequence of $\eta_{(1)} \leq \cdots \leq \eta_{(C)}$, which is $-\eta_{(C)} \leq \cdots \leq -\eta_{(1)}$, results in the same 1-dimensional space (up to an irrelevant sign). This means we need to study the remaining $C!/2$ possible convex polyhedrons.

As noted in the proof of our theorem above, when the dimensionality of the space spanned by the (whitened) class means is strictly smaller than $C-1$ (i.e., $dim(span\{\hat{\mu}_1, \ldots, \hat{\mu}_C\}) = rank(\hat{\mathbf{S}}_B) < C-1$), then not all the sequences are possible. This can be easily detected, because in such cases

the convex region $\mathcal{A}$ is the origin. Therefore, in general, we will need to solve a smaller number of convex problems with $C!/2$ being the upper limit.

We also need to restrict our search space with those vectors that have a unit length only, i.e., $\mathbf{v}^T\mathbf{v} = 1$. Although this constraint specifies a non-convex region (a hypersphere or S^{d-1}) we can relax this to be a convex region by using the following reasoning. For a given unit vector $\mathbf{v}$, any $\alpha\mathbf{v}$ with $\alpha < 1$ will produce a larger Bayes error, i.e., $g(\alpha\mathbf{v}) > g(\mathbf{v})$. This means that we can consider the intersection of the unit ball and $\mathcal{A}$ as our search space $\mathcal{B}$, i.e., $\mathcal{B} = \mathcal{A} \bigcap \mathbb{B}^p$, where $\mathbb{B}^p$ is the p-dimensional unit ball, defined as the set of all vectors $\mathbf{x}$ for which $\|\mathbf{x}\| \leq 1$, and $\|\mathbf{x}\|$ is the 2-norm length of the vector $\mathbf{x}$. This is possible because the solution $\mathbf{v} \in \mathcal{B}$ that minimizes our criterion will have to be a unit vector. It is important to note that now the search region $\mathcal{B}$ is a convex region, which again permits the use of any convex optimization algorithm.

Our algorithm can thus be summarized as follows. First, find the set Q of possible orderings of the whitened class means. This is easily achieved by selecting all those sequences for which $\mathcal{A}$ is larger than the origin. Second, for each ordering q_i in Q find that $\mathbf{v}_i \in \mathcal{B}$ which minimizes the Bayes error by using a convex optimization algorithm (e.g., the active set sequential quadratic programming method [7]). Finally, the optimal solution $\mathbf{v}$ to our problem is given by

$$\mathbf{v} = \arg\min_{\mathbf{v}_i} g(\mathbf{v}_i). \tag{4}$$

An example application of this algorithm was shown in Fig. 1, where in (a) we showed a six class problem and in (b) the solution obtained by the algorithm described in this section (which is the Bayes optimal). Our solution is also compared to that given by Fisher's LDA, which was shown in Fig. 1(c). Further comparisons to other methods and additional examples are given in Appendix B.

Finally, note that since the nearest mean classification rule is optimal for homoscedastic Gaussian distributions, this should be the classifier of our choice in the testing stage.

4 Extensions and Approximations

In our previous section, we presented an algorithm to find that 1-dimensional space $\mathbf{v} \in \mathbb{R}^p$ where the Bayes error is minimized whenever the class densities are homoscedastic. We now present extensions that will allow us to use our result in a large number of real applications.

4.1 Nonlinear Bayes solution

For practical applications, it is convenient to relax the assumption of homoscedasticity to one which permits us to work with the nonlinear decision boundaries resulting from heteroscedastic distributions. We can achieve this by means of the well-known kernel trick.

As in LDA, the criterion of Kernel LDA (KLDA, sometimes called Generalized LDA) [15, 2] will also aim at the least squares solution maximizing the distance between the projected class means. Once more, this solution will be biased toward those class means that are farthest apart. By using our main result, however, we can resolve this problem.

Let μ_i^ψ represent the mean of class i in the high dimensional space $\mathcal{F}$ obtained with the mapping function $\psi(.)$. Each class mean is given by $\mu_i^\psi = \frac{1}{n_i}\sum_{j=1}^{n_i}\psi(\mathbf{x}_j^i)$, with n_i being the number of samples in class i and $\mathbf{x}_j^i$ is the j^{th} sample in class i. The average within class scatter matrix is

$$\bar{\Sigma}^\Psi = \frac{1}{C}\sum_{i=1}^{C}\sum_{j=1}^{n_i}(\psi(\mathbf{x}_j^i) - \mu_i^\psi)(\psi(\mathbf{x}_j^i) - \mu_i^\psi)^T.$$

This equation can be rewritten as follows

$$\begin{aligned}\bar{\Sigma}^{\Psi} &= \frac{1}{C}\sum_{i=1}^{C}\Psi(\mathbf{X}^i)\Psi(\mathbf{X}^i)^T - \sum_{j=1}^{n_i}\psi(\mathbf{x}_j^i)\mu_i^{\psi T} - \mu_i^{\psi}\sum_{j=1}^{n_i}\psi(\mathbf{x}_j^i)^T + \mu_i^{\psi}\mu_i^{\psi T} \\ &= \frac{1}{C}\sum_{i=1}^{C}\Psi(\mathbf{X}^i)\Psi(\mathbf{X}^i)^T - \Psi(\mathbf{X}^i)\mathbf{1}_{n_i}\Psi(\mathbf{X}^i)^T \\ &= \frac{1}{C}\sum_{i=1}^{C}\Psi(\mathbf{X}^i)(\mathbf{I}-\mathbf{1}_{n_i})\Psi(\mathbf{X}^i)^T,\end{aligned}$$

where $\Psi(\mathbf{X}^i) = \big(\psi(\mathbf{x}_1^i),\ldots,\psi(\mathbf{x}_{n_i}^i)\big)$ is the matrix of sample vectors in $\mathcal{F}$, $\mathbf{I}$ is the identity matrix and $\mathbf{1}_{n_i}$ is $n_i \times n_i$ matrix with all elements equal to $1/n_i$.

The eigenvectors $\mathbf{W}^{\Psi}$ and the eigenvalues Λ^{Ψ} of $\bar{\Sigma}^{\Psi}$ can be calculated using the kernel trick. As we know $\mathbf{W}^{\Psi T}\bar{\Sigma}^{\Psi}\mathbf{W}^{\Psi} = \Lambda^{\Psi}$, where $\mathbf{W}^{\Psi}$ is defined as a linear combination of the samples, since $\mathbf{W}^{\Psi}$ is in the span of $\Psi(\mathbf{X})$, which can be restated as $\mathbf{W}^{\Psi} = \Psi(\mathbf{X})\Gamma$, with $\Psi(\mathbf{X}) = (\psi(\mathbf{x}_1),\ldots,\psi(\mathbf{x}_n))$. Letting $\mathbf{N} = \frac{1}{C}\sum_{i=1}^{C}\mathbf{K}_i(\mathbf{I}-\mathbf{1}_{n_i})\mathbf{K}_i^T$, with $\mathbf{K}_i = \Psi(\mathbf{X})^T\Psi(\mathbf{X}^i)$, allows us to write

$$\Gamma^T\mathbf{N}\Gamma = \Lambda^{\Psi}, \tag{5}$$

where Γ is the coefficient matrix. With this result, we can compute the whitened class means as

$$\begin{aligned}\widehat{\mu}_i^{\psi} &= \Lambda^{\Psi^{-1/2}}\mathbf{W}^{\Psi T}\mu_i^{\psi} = \Lambda^{\Psi^{-1/2}}\Gamma^T\Psi(\mathbf{X})^T\frac{1}{n_i}\sum_{j=1}^{n_i}\psi(\mathbf{x}_j^i) \\ &= \frac{1}{n_i}\Lambda^{\Psi^{-1/2}}\Gamma^T\mathbf{K}_i\mathbf{1},\end{aligned} \tag{6}$$

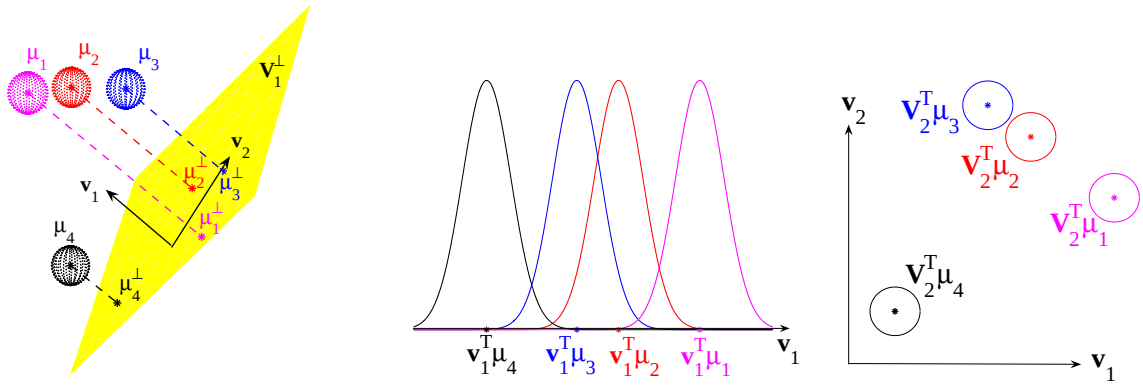
where $\mathbf{1}$ is a $n_i \times 1$ vector of ones. For a given sequence $q_j \in Q$ and the whitened class means, we can obtain our solution by solving for $g(\mathbf{w}_j) = 2/C\sum_{i=1}^{C-1}\Phi(-\mathbf{w}_j^T(\hat{\mu}_{i+1}^{\psi} - \hat{\mu}_i^{\psi})/2)$, with the $C-1$ constraints $\mathbf{w}_j^T(\hat{\mu}_{i+1}^{\psi} - \hat{\mu}_i^{\psi}) \geq 0$. Following our main result, the Bayes optimal one-dimensional subspace is

$$\mathbf{w} = \arg\min_{\mathbf{w}_j} g(\mathbf{w}_j). \tag{7}$$

New samples can then be projected into the reduced space with $\Psi(\mathbf{X})\Gamma\Lambda^{\Psi^{-1/2}}\mathbf{w}$.

4.2 Optimal d-dimensional subspace

To find a subspace solution of more than one dimension, we recursively apply our algorithm to the null space of the previously obtained subspace. This is illustrated in Fig. 4. In this example we have four classes. After applying the algorithm described in Section 3, one obtains the first subspace solution, which is labeled $\mathbf{v}_1$ in Fig. 4(a). The resulting projection of the original distributions onto $\mathbf{v}_1$ is shown in Fig. 4(b). The null space of this first solution is shown as a colored hyperplane in Fig. 4(a) and denoted $\mathbf{V}_1^{\perp}$. Now, we can re-apply the same algorithm within this null space $\mathbf{V}_1^{\perp}$, obtaining the second optimal dimension $\mathbf{v}_2$. To do this, we will first need to project the class means onto $\mathbf{V}_1^{\perp}$ and then calculate the next solution on this null space. The 2-dimensional subspace where the Bayes error is minimized is then given by the projection matrix $\mathbf{V}_2 = (\mathbf{v}_1, \mathbf{v}_2)$, with the usual constraint $\mathbf{v}_1^T\mathbf{v}_2 = 0$. The resulting 2-dimensional projection of the original distributions is shown in Fig. 4(c). More generally, our algorithm can be recursively applied to find that d-dimensional



(a) (b) (c)

Figure 4: *Our algorithm can be recursively applied to obtain subspaces of any dimensionality. In (a) we show the first one-dimensional subspace* $\mathbf{v}_1$ *where the four distributions generate the smallest Bayes error. The resulting projection is shown in (b). This result is obtained using Eq. (4). The colored hyperplane* $\mathbf{V}_1^\perp$ *shown in (a) is the null space of our solution* $\mathbf{v}_1$*. We can now project the class means onto this null space (denoted* $\mu_i^\perp$*) and there calculate the second solution* $\mathbf{v}_2$ *using the same algorithm, which is the Bayes optimal in* $\mathbf{V}_1^\perp$*. This generates the 2-dimensional subspace solution given by* $\mathbf{V}_2 = (\mathbf{v}_1, \mathbf{v}_2)$*. (c) The projection of the original class distributions onto* $\mathbf{V}_2$*.*

Algorithm 1 d-dimensional subspace

Let $\hat{\mu}_i = \bar{\Lambda}^{-1/2}\bar{\mathbf{V}}^T\mu_i$ be the whitened mean locations. Here, μ_i is the mean of class i, and $\bar{\Lambda}$ and $\bar{\mathbf{V}}$ are the eigenvector and eigenvalue matrices of the average class covariance matrices defined as $\bar{\Sigma} = C^{-1}\sum_{i=1}^{C}\Sigma_i$.
Calculate $\mathbf{v}_1$ using Eq.(4) and the mean locations $\hat{\mu}_i$. Set $\mathbf{V}_1 = \mathbf{v}_1$.
for $k = 2$ to d **do**
  Project $\hat{\mu}_i$ to the null space $\mathbf{V}_{k-1}^\perp$ of the current solution $\mathbf{V}_{k-1}$, i.e., $\tilde{\mu}_i = \mathbf{V}_{k-1}^{\perp T}\hat{\mu}_i$.
  Calculate $\mathbf{v}_k$ using Eq.(4) with $\tilde{\mu}_i$.
  $\mathbf{V}_k = \left(\mathbf{V}_{k-1}, \mathbf{V}_{k-1}^\perp\mathbf{v}_k\right)$.
end for

solution from any p-dimensional space, with $d \leq \min(p, C-1)$. This is summarized in Algorithm 1.

Similarly, we can extend (7) to find d orthogonal dimensions in the kernel space. This is shown in Algorithm 2.

4.3 Improving on other algorithms

An important application of our result is in improving the outcomes given by other algorithms. To see this, note that each method (e.g., LDA, aPAC, and FLDA) will produce a subspace result defined by a projection vector $\mathbf{V}$. In this subspace, we will have a sequence of the projected means. We can now use our approach to find the Bayes optimal solution for this given sequence.

This result can be implemented as follows. We first use the algorithm of our choice (Algorithm A, e.g., LDA) to find that 1-dimensional space where the data is "best" classified according to its criterion. This solution is given by the projection vector $\mathbf{v}_1$. We use Eq. (4) to find the optimal first dimension $\mathbf{v}_1^*$ by applying the algorithm only to the sequence q_i given by $\mathbf{v}_1$. Then, we project the class means to the null space of $\mathbf{v}_1^*$ and use Algorithm A to find the second dimension $\mathbf{v}_2$ and use the

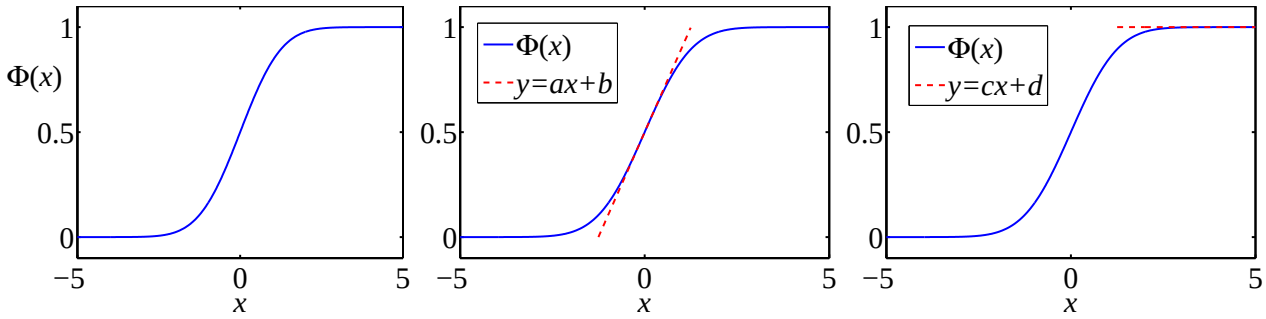
Algorithm 2 Kernel d-dimensional subspace

Let $\hat{\mu}_i^{\psi}$ be the whitened class mean locations in the kernel space.
Calculate $\mathbf{w}_1$ using Eq. (7) and the mean locations $\hat{\mu}_i^{\psi}$. Set $\mathbf{W}_1 = \mathbf{w}_1$.
for $k = 2$ to d **do**
  Project $\hat{\mu}_i^{\psi}$ to the null space $\mathbf{W}_{k-1}^{\perp}$ of the current solution $\mathbf{W}_{k-1}$, i.e., $\tilde{\mu}_i^{\psi} = \mathbf{W}_{k-1}^{\perp^T}\hat{\mu}_i^{\psi}$.
  Calculate $\mathbf{w}_k$ using Eq. (7) with $\tilde{\mu}_i^{\psi}$.
  $\mathbf{W}_k = \left(\mathbf{W}_{k-1}, \mathbf{W}_{k-1}^{\perp}\mathbf{w}_k\right)$.
end for

approach described in Section 3 to obtain $\mathbf{v}_2^*$ by considering the sequence given by $\mathbf{v}_2$. Repeat this process until selecting the d desired dimensions, $\mathbf{V}_d^* = (\mathbf{v}_1^*, \ldots, \mathbf{v}_d^*)$. We will refer to this extension of any Algorithm A as Algorithm A$_{opt}$, where *opt* is short for optimized.

4.4 Linear approximation

As already mentioned earlier, the number of possible ordered sequences is upper-bounded by $C!/2$. This is so at each iteration of the algorithm. The exact number of possible sequences at the i^{th} iteration of the algorithm, h_i, can be easily computed in practice for any given particular data-set. This is given by the number of sequences where (3) has a common solution larger than the origin. The complexity of our algorithm is then given by $h = 1/2 \sum_{i=1}^{d} h_i!$. When h is large, approximations are necessary.



(a) (b) (c)

Figure 5: *(a) The y-axis corresponds to the values of* $\Phi(x)$ *for the corresponding values shown in the x-axis. (b) Plotted here is the linear approximation of* $\Phi(x)$ *for values of x close to zero. This linear approximation is the tangent to* $\Phi(0)$*, i.e.,* $y = ax+b$ *with* $a = 1/\sqrt{2\pi}$ *and* $b = 1/2$*. (c) Linear approximation for values of* $x > \sqrt{\pi/2}$*. This second approximation is given by* $y = cx + d$ *for* $x > 0$*. Note that because* $\Phi(x)$ *increases very slowly when x is larger than* $\sqrt{\pi/2}$*, c should be close to zero, and d should be close to* 1*.*

Recall that the Bayes error function for an ordered sequence $\eta_{(i)}$ can be written as in (2). The complexity of this function comes from the nonlinearity of the cumulative distribution function $\Phi(x)$, since this weights each of the possible ordered sequences differently depending on the distances between their projected class means. This is illustrated in Fig. 5(a), where the x-axis represents the distance between projected means, and the y-axis the corresponding weights given by $\Phi(x)$. Note that at first these weights increase quite fast, but latter saturate and remain almost constant. These two behaviors can be easily approximated with linear functions as shown in Fig. 5(b-c). The first of these approximations is very useful, because if the distances between class means in the original space $\mathbb{R}^p$ are small, then their distances will also be small in *any* reduced space $\mathbf{V}_d$. This

means we can substitute (2) by the following linear approximation

$$\frac{1}{C}\sum_{i=1}^{C-1}\left(a\left(\eta_{(i)}-\eta_{(i+1)}\right)+b\right).$$

Since $a > 0$ and b are constants, minimizing the above equation is equivalent to minimizing $\left(\eta_{(1)}-\eta_{(C)}\right)$. The great advantage of this linear approximation is that now this function solely depends on the first and last elements of the sequence. Rewriting this equation as $-\mathbf{v}^T\hat{\mu}_{(1),(C)}$, where $\eta_{(i)}=\mathbf{v}^T\hat{\mu}_{(i)}$, shows that the minimum is achieved when $\mathbf{v}=\frac{\hat{\mu}_{(1),(C)}}{\|\hat{\mu}_{(1),(C)}\|}$. Therefore, we see that for any possible ordered sequences of projected means, the linear approximation to the Bayes error is

$$-\frac{a\|\hat{\mu}_{(1),(C)}\|+(C-1)b}{C}. \tag{8}$$

Since our goal is to find that sequence (or equivalently, that $\mathbf{v}$) where this error is minimized, our algorithm now reduces to finding which of all possible ordered sequences minimizes (8), which is given by that pair of class means that are farthest apart in $\mathbb{R}^p$. More formally, the solution given by the linear approximation just described is

$$\mathbf{v}_{app}=\frac{\hat{\mu}_{i,j}}{\|\hat{\mu}_{i,j}\|}, \tag{9}$$

where i and j are given by

$$\arg\max_{i,j}\|\hat{\mu}_{i,j}\|. \tag{10}$$

This algorithm requires that we compute $C(C-1)/2$ distances in $\mathbb{R}^r$, $r=rank(\hat{\mathbf{S}}_B)$, which means that its complexity is of the order of rC^2.

Again, we can use this algorithm to find any d-dimensional subspace by recursively computing equations (9)-(10) in the null space of the previous solution. This is summarized in Algorithm 3.

Algorithm 3 Linear Approximation

Let $\hat{\mu}_i$ be the whitened means.
$\mathbf{V}_{app_1}=\mathbf{v}_{app_1}=\frac{\hat{\mu}_{i,j}}{\|\hat{\mu}_{i,j}\|}$, where i and j are given by $\arg\max_{i,j}\|\hat{\mu}_{i,j}\|$.
for $k=2$ to d **do**
  Project $\hat{\mu}_i$ to the null space $\mathbf{V}^{\perp}_{app_{k-1}}$ of the current solution, i.e., $\tilde{\mu}_i=\mathbf{V}^{\perp^T}_{app_{k-1}}\hat{\mu}_i$.
  Calculate $\mathbf{v}_{app_k}=\frac{\tilde{\mu}_{i,j}}{\|\tilde{\mu}_{i,j}\|}$, where i and j are given by $\arg\max_{i,j}\|\tilde{\mu}_{i,j}\|$.
  $\mathbf{V}_{app_k}=\left(\mathbf{V}_{app_{k-1}},\mathbf{V}^{\perp}_{app_{k-1}}\mathbf{v}_{app_k}\right)$.
end for

However, this algorithm is only guaranteed to find a solution close to the optimal one when the class means are all close to each other. Ideally, we would like to have a criterion which can be used to determine when this happens. This can be readily done by setting a threshold x_0 for which the linear approximation is no longer a close match of (2). A useful value for x_0 is 1, because the acceleration (second derivative) of the slope of $\Phi(.)$ is 0 at such a point. This implies the largest distance between class means cannot exceed 2 if we are to successfully apply this algorithm.

Shall one insist in applying this algorithm in a problem where the above given criterion does not hold, the solution of Algorithm 3 can be further tuned by applying the method described in Section 3 for the given sequence of class means seen in $\mathbf{V}_{app}$. Following our notation, this will result in Algorithm 3_{opt}.

We now go back to Fig. 5(c) to realize that a similar argument can be build to construct an algorithm applicable when all between-class distances are large. Recall that in our previous case, we could use an approximation because we knew that small between-class distances in the original space $\mathbb{R}^p$ cannot become large distance in the reduced space $\mathbb{R}^d$. Unfortunately, this is not true for large distances, because large distances in the original space can become small in the reduced space. This means that we can only construct an approximation when its solution provides a subspace where the original (large) distances do not go below a known threshold. From Fig. 5(c) we see that the cdf can be approximated with $\Phi(x) \approx cx + d$. As we did above, we can now rewrite the criterion to be minimized as $\left(\eta_{(1)} - \eta_{(C)}\right)$, which is the same solution given for the small between-class distance case, i.e., $\mathbf{v}_{app} = \hat{\mu}_{i,j}/\|\hat{\mu}_{i,j}\|$, where i and j are given by $\arg\max_{i,j} \|\hat{\mu}_{i,j}\|$. Nonetheless, we still need to guarantee that all the between-class distances remain above a threshold in $\mathbf{v}_{app}$. A convenient value for this threshold is 3, because after it the cdf contains $\sim$99.7% of the probability, which implies that $\Phi(x)$ will be almost flat for any $x > 3$. Therefore, from Eq. (2), we know that our solution will be a good approximation if all $\|\eta_{(i)} - \eta_{(i+1)}\| \geq 6$.

One could also argue that because the approximations derived above are identical, one could use these in other more generic cases (e.g., when we have large and small distances). Unfortunately, this would be misleading because our solution would only hold if the between class differences remain below and above the specified thresholds in the reduced space. If they do not, then we do not have a mechanism to correct our solution. This is in fact what Fisher-Rao's LDA does. Recall from Section 2 that LDA is biased toward the largest between-class distance (similar to our approximation $\mathbf{v}_{app}$). When the thresholds defined above hold in the reduced space, LDA provides a reasonable solution; otherwise, its performance is unknown.

5 Experimental Results

In Figs. 1 and 4, we showed results obtained using Algorithm 1 on synthetic data. Results obtained using other algorithms on these examples are in Appendix B, where we further develop on the use of synthetic data. In this section, we will report on a statistical analysis of an additional synthetic-data-set and then focus our attention to the use of real data. For comparison purposes, in all algorithms, we assume equal priors. Also, recall that to classify new samples, we will use the nearest mean rule, which provides optimal classification under homoscedasticity.

5.1 A study of the linear approximation

We want to study how accurate the results obtained with our linear approximation is to the optimal solution described in Section 3. In particular, we are interested in determining whether these results compared to those of LDA or, as predicted earlier, are generally associated to a smaller classification error.

In our study, we randomly generate six (class) mean locations from within a ball of radius r, with $r = \{1, 2, \ldots, 10\}$. This variance can correspond to the class distribution or the data noise. We consider the random generation process to be uniformly distributed. The resulting mean locations are assigned an identity matrix as covariance matrix, which is the same as assuming we are in the whitened space. This process is repeated 1,000 times and Algorithms 1 and 3 as well as LDA are used to obtain the 1-dimensional subspace solutions, with dimensionality of the original space $p = \{2, 3, 4, 5\}$. For each of the methods, we calculate the average and standard deviation of the Bayes error in the reduced 1-dimensional space for the above specified values of r and p. The Bayes error is calculated using (2). The results are shown in Fig. 6(a-d).

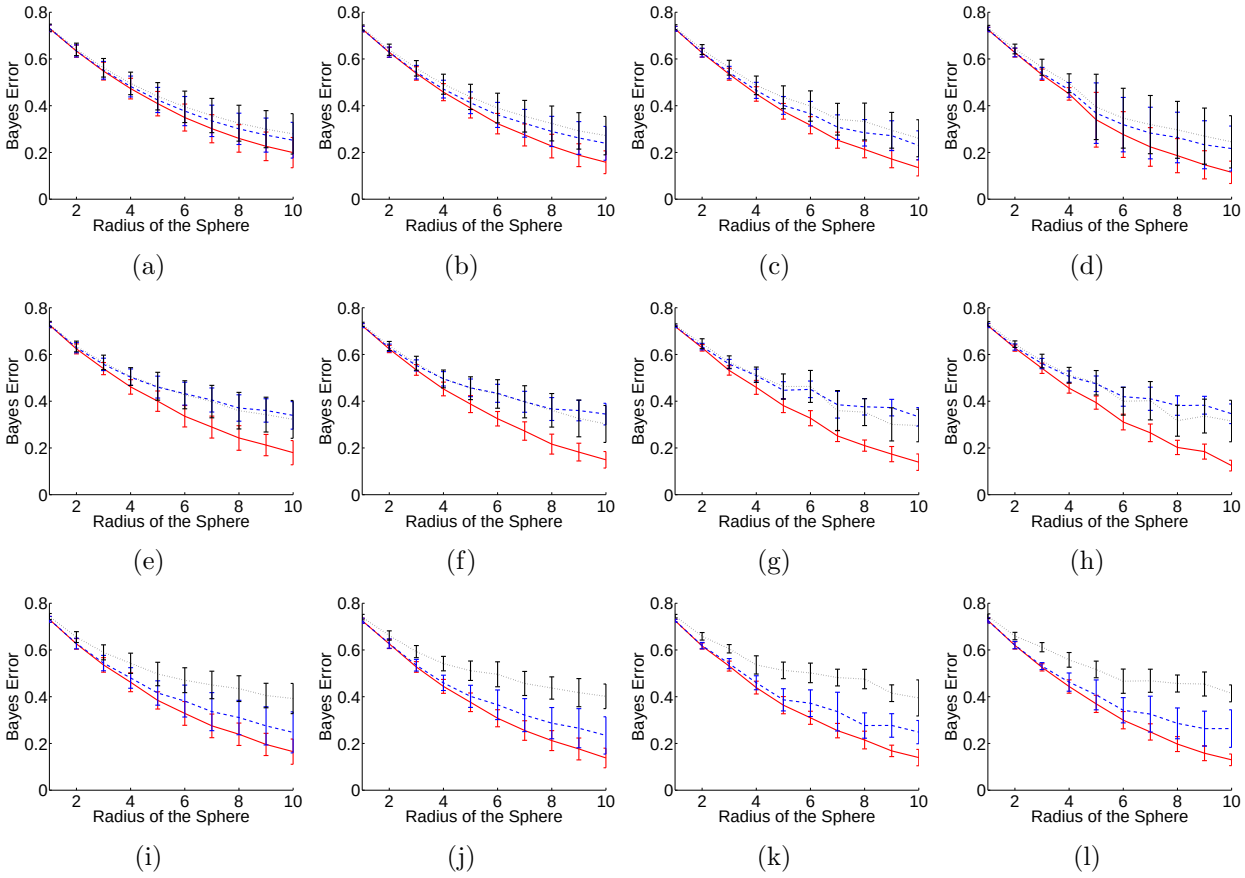


Figure 6: *(a-d) Average and standard deviations of the Bayes error calculated with (2) on the subspace generated by Algorithms 1 (solid lines), 3 (dashed lines) and LDA (dotted lines) over a total of 1,000 sets with 6 homoscedastic class distributions. In each trial, the covariance matrices are set to be equal to the identity matrix and the mean locations are randomly drawn from a ball of radius r specified in the x axis of each plot. (e-h) Average and standard deviations for the 100 worse runs of Algorithm 3. (i-l) Average and standard deviation of the 100 worse cases for LDA. The dimensionality of the original space p is: (a,e,i) 2, (b,f,j) 3, (c,g,k) 4, (d,h,l) 5.*

As we can see in these results, Algorithm 3 performs slightly better than LDA. Next, we would like to know the reasoning behind this preference. To test this, we first selected the 100 runs (from the total of 1,000 mentioned above) where Algorithm 3 performs most poorly when compared to Algorithm 1. We wanted to know whether in these cases LDA would be preferred over our approximation. The average and standard deviation of these 100 results are in Fig. 6(e-h). It is noticed that the worse cases of Algorithm 3 are no worse than LDA. The obvious next question is to see what happens when LDA performs poorly. To see this, we selected the 100 cases where LDA performed worse when compared to Algorithm 1. The average and standard deviations over these 100 cases for each of the three algorithms are in Fig. 6(i-l). It is now clear that in those cases where LDA performs worse, our linear approximation does better. This is then the explanation for the superiority of the new approximation.

We also see that the results of Algorithm 3 generally degrade for values of the radius larger than 6. This is because, in such cases, the distance between class means can easily be more than 4, which is the upper-limit mentioned earlier for the algorithm to work well. The results of LDA start to deviate from those of the optimal solution earlier. This is because LDA is driven by all the class means, which (as demonstrated in Section 4.4) is not always adequate.

5.2 Object and face recognition

A classical problem in computer vision is object categorization. Here, images of objects have to be classified according to a set of pre-defined categories, e.g., cows versus cars. To test our algorithm in this scenario, we used the ETH-80 database [8]. This database includes the images of eight categories: apples, cars, cows, cups, dogs, horses, pears and tomatoes. Each of these categories is represented by the images of ten objects (e.g., ten cars) photographed from a total of 41 orientations. This means that we have a total of 410 images per category. The reason for selecting this database comes from the known fact that this is a difficult data-set for classical discriminant analysis algorithms [13].

A common feature space in object recognition is one that uses the derivatives of the Gaussian filter. In our experiment, we consider the first derivative about the x and y axes for a total of three different scales, i.e., $\sigma = \{1, 2, 4\}$, where σ is the variance of the Gaussian filter. For computational simplicity, it is convenient to represent the responses of these filters in the form of a histogram. Our histogram representation has a total of 32 entries which are sampled across all filter responses. We refer to this feature representation as *GaussXY*.

As an alternative, we also experimented with the use of the magnitude of the gradient and the Laplacian operator, which generate rotation-invariant information. As above, the gradient and Laplacian are computed using three scales, i.e., $\sigma = \{1, 2, 4\}$. To reduce the number of features obtained, we also use the histogram representation described above (which results in feature vectors of 32 dimensions). This second representation will be refereed to as *MagLap*.

These two image representations are tested using the leave-one-object-out strategy, where (at each iteration) we leave one of the sample objects out for testing and use the remaining for training. This is repeated (iterated) for each of the possible objects that we can left out. Table 1 show the average classification rates for all the algorithms introduced earlier and an implementation of the PCA-LDA approach. In the PCA-LDA algorithm, PCA is first used to reduce the dimensionality of the original space to facilitate LDA's task. These results are shown for the dimensions $d = \{1, 2, 3, 4, 5, 6, 7\}$.

We see that, as expected, Algorithm 2 provides the best classification result. This difference is especially marked in the lowest-dimensional spaces. In fact a single dimension suffices to obtain classification results above 80% (which is better than what all the other methods can achieve even when $d = 5$). This is very significative, because it allows us to define a compact feature space that

	GausXY							MagLap						
Dimensionality (d)	1	2	3	4	5	6	7	1	2	3	4	5	6	7
Algorithm 1	53.38	66.25	71.46	72.29	76.65	77.87	81.68	57.53	67.04	72.23	72.80	74.02	76.62	78.51
Algorithm 2	**81.62**	**84.70**	**86.01**	**88.29**	**89.21**	**89.48**	**92.04**	**84.21**	**86.43**	**88.75**	**89.05**	**91.01**	**91.86**	**92.47**
Algorithm 3	52.65	63.84	73.02	72.20	77.44	80	81.68	53.69	65.03	74.15	73.41	78.69	78.23	78.50
Algorithm 3_{opt}	53.38	67.99	70.55	72.04	76.40	79.63	81.68	54.15	69.24	71.80	73.29	77.65	77.13	78.50
LDA	46.16	61.10	68.99	69.51	73.32	78.08	81.68	46.89	62.35	70.24	70.73	74.51	76.98	78.50
LDA_{opt}	53.54	68.23	70.98	73.20	75.15	77.10	81.67	54.30	69.21	72.23	74.45	76.40	76.76	78.50
$aPAC$	44.70	62.59	68.81	70.73	74.45	80.15	81.68	46.62	60.98	69.85	72.68	75.55	77.99	78.50
$aPAC_{opt}$	53.54	68.93	71.31	73.72	76.52	78.51	81.68	57.74	68.54	71.07	73.11	75.88	77.23	78.50
$FLDA$	43.48	67.35	73.38	78.05	80.37	81.16	81.68	43.08	66.52	71.4	77.1	79.21	78.51	78.51
$FLDA_{opt}$	47.41	66.37	72.96	75.58	76.71	79.33	81.68	47.16	66.43	71.55	73.29	74.24	77.2	78.51
$DFLDA$	27.84	44.36	54.73	62.23	62.56	63.08	63.29	35.21	43.48	53.05	63.81	68.14	69.18	69.33
$DFLDA_{opt}$	37.26	56.04	60.21	63.2	62.35	62.9	63.29	42.74	56.4	61.71	65.34	66.49	69.36	69.33
PCA-LDA	46.19	62.93	63.08	64.27	64.42	66.25	66.07	34.18	56.34	61.65	64.6	64.51	62.68	67.1
PCA-LDA_{opt}	46.34	62.96	63.81	64.33	64.57	66.77	66.07	33.35	57.41	60.43	64.45	64.18	63.29	67.13
NDA	45.55	60.64	69.57	70.52	72.07	74.63	77.9	48.6	62.07	67.9	70.82	71.01	73.35	76.16
NDA_{opt}	52.26	68.2	70.64	75.52	76.25	78.29	82.04	56.95	70.27	71.95	73.32	75.4	77.71	78.63
$KLDA$	62.80	74.18	73.32	78.69	79.97	85.85	**91.83**	63.20	69.51	73.81	80.18	81.10	88.14	**92.44**

Table 1: *Average classification rates for the ETH database with* GausXY *and* MagLap *feature representations. The kernel function used in Algorithm 2 and KLDA is* $\psi(\mathbf{x})^T\psi(\mathbf{y}) = \left(\mathbf{x}^T\mathbf{y}\right)^k$*, where* k *is optimized using the leave-one-object-out strategy. This process resulted in* $k = 9$ *and* 8 *in Algorithm 2 with* GaussXY *and* MagLap*, respectively, and* $k = 3$ *for KLDA in both representations.*

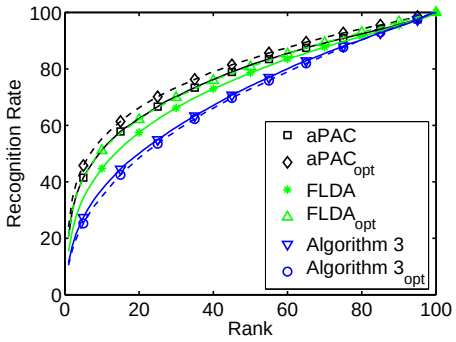


Figure 7: *Plotted here are the cumulative match scores with* $d = 8$ *for the FRGC v2 database. The algorithms tested were* $aPAC$*,* $aPAC_{opt}$*,* $FLDA$*,* $FLDA_{opt}$*, Algorithm 3, and Algorithm* 3_{opt}*.*

will rarely accept false positives. We also note that the optimization mechanism described in this paper helps boost the classification results obtained with off-the-shelf algorithms – especially when very low-dimensional feature representations are needed.

Thus far, we have seen the usefulness of our main result and its kernel extension in object categorization. We have also demonstrated the uses of the optimization criterion of Section 4.3. That is, in general, Algorithm A_{opt} was able to boost the classification results of Algorithm A. We would now like to further study this point in a problem that involves a larger number of classes. To this end, we have used the FRGC (Face Recognition Grand Challenge) version 2 dataset [17]. This database has an additional challenge given by the limited number of samples per class. This makes the problem more difficult, because we now need to learn from underrepresented distributions. In our test, we used all the images in the target set, corresponding to 466 individuals. This provides a total of $16,028$ face images. All images were aligned, cropped, and resized to an image size of 150×130 pixels. The images were also normalized to vectors with zero-mean and unit-variance. A set of $1,024$ training images are used to compute a lower-dimensional PCA space where subsequent classification is carried out. We ran a leave 100-out cross validation test in this subspace. This means, 100 samples from the data-set are left out for testing. This process is repeated a total of 200 times. The average cumulative match scores for the dimensionality of 8 is shown in Fig. 7. These results are obtained in the Kernel space defined by a radial basis function $\psi(\mathbf{x})^T\psi(\mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right)$ with $\sigma = 1$.

Dimensionality (d)	1	2	3	4	5	6
Algorithm 1	**71.24**	**80.86**	**85.24**	88.05	89.95	90.47
Algorithm 2	**71.14**	**80.05**	**86.05**	87.33	88.00	87.52
Algorithm 3	62.62	70.71	83.52	88.1	89.86	90.47
Algorithm 3_{opt}	**71.24**	73.14	83.86	88.81	90.71	90.47
LDA	51.00	66.09	82.14	87.95	89.81	90.47
LDA_{opt}	**71.24**	73.91	84.00	88.48	90.38	90.47
$aPAC$	64.95	66.05	82.76	87.48	89.90	90.47
$aPAC_{opt}$	**71.23**	**80.85**	**85.23**	87.71	89.61	90.47
$FLDA$	55.14	81.05	85.33	90.1	90.57	90.47
$FLDA_{opt}$	65.48	83.76	87.48	88.29	90.29	90.47
$DFLDA$	54.9	60.33	74.48	79.86	79.76	79.71
$DFLDA_{opt}$	58.86	67.05	73.14	78.76	79.67	79.71
$PCA-LDA$	27.95	49.29	67.71	79.19	81.29	82.38
$PCA-LDA_{opt}$	64.19	68.1	78.81	80.05	81.81	82.38
NDA	65.76	65.1	82.57	87.9	88.43	88.67
NDA_{opt}	**71.24**	73.05	83.81	87.38	90.62	90.76
$KLDA$	64.33	73.81	84.48	85.71	86.67	87.52

Table 2: *Successful classification rates on the image segmentation data-set: Shown here is the probability of correct classification (in percentages) for each of the algorithms and their optimized versions. The results are shown for a set of possible low-dimensional spaces. The kernel function is* $\psi(\mathbf{x})^T\psi(\mathbf{y}) = \left(\mathbf{x}^T\mathbf{y}\right)^k$*, where* k *is optimized using 10-fold cross validation over the training set. This provides* $k = 2$ *for both Algorithm 2 and KLDA.*

5.3 UCI repository

Image segmentation: Our next test uses the image segmentation data-set of the UCI machine learning repository [16]. The samples in this set belong to one of the following seven classes: brickface, sky, foliage, cement, window, path and grass. The data-set is divided into a training set of 210 images (30 samples per class) and a 2,100 testing set (300 samples per class). The feature space is constructed using 18 features extracted from 3×3 image patches.[2] These represent: the pixel location of the patch's middle point, the number of lines in the region, mean and standard deviations of the contrast of horizontally and vertically adjacent pixels, the intensity information of the RGB color space, a 3D nonlinear transformation of the color space, and the hue and saturation means.

Table 2 summarizes the successful classification rate on the testing set for the following values of the dimensionality $d = \{1, 2, 3, 4, 5, 6\}$. We have bolded those values which provide a reasonable improvement over the others. From this, we see that the Bayes solutions provided by Algorithm 1 and 2 always are among the best and the only ones to be consistently among the top in the lowest-dimensional representations. Shall one want to find that smallest space where the data is best separated, the use of these algorithms becomes imperative. To further illustrate this, we provide 2- and 1-dimensional plots of the data in Appendix B (which is available in the online Supplementary Documentation).

We also see that our optimization procedure does indeed help improve the results obtained with different algorithms (e.g., LDA, aPAC and FLDA). In many cases, this improvement is quite remarkable.

Satellite imagery: Our final test uses the Landsat data of the Statlog project defined in [16]. This data-set has a total of six classes, with samples described by 36 features. These features correspond to 4 spectral band values of 3×3 satellite image patches. The set includes 4,435 training samples and 2,000 testing samples.

Table 3 summarizes the classification rates obtained with each of the algorithms tested for the

[2]The original data-set contains 19 features. We have however eliminated feature number 3 after realizing that all the values in that dimension were the same.

Dimensionality (d)	1	2	3	4	5
Algorithm 1	69.65	80.65	82.80	82.55	83.15
Algorithm 2	**75.50**	**82.60**	82.85	**83.50**	**84.60**
Algorithm 3	58.80	77.95	82.15	82.55	83.15
Algorithm 3_{opt}	61.60	77.55	79.50	82.60	83.15
LDA	62.35	74.10	82.50	82.70	83.15
LDA_{opt}	69.70	80.65	82.75	82.55	83.15
$aPAC$	65.25	78.65	82.75	82.60	83.15
$aPAC_{opt}$	58.90	78.40	82.45	82.55	83.15
$FLDA$	60.1	72.65	80.9	82.65	83.15
$FLDA_{opt}$	67.75	81.8	82.45	82.5	83.15
$DFLDA$	64.6	76.15	81.6	82.85	83.05
$DFLDA_{opt}$	66.85	77.3	82.25	82.8	83.05
$PCA-LDA$	39.15	60.75	78.45	78.5	78.35
$PCA-LDA_{opt}$	43	73.9	78.2	78.2	78.35
NDA	53.75	78.2	82.7	**83.1**	82.95
NDA_{opt}	65.85	80.55	82.75	**83.45**	83.4
$KLDA$	57.45	78.50	78.60	77.65	78.45

Table 3: *Successful classification rates on the LandSat data-set. The kernel function used is* $\psi(\mathbf{x})^T\psi(\mathbf{y}) = \left(\mathbf{x}^T\mathbf{y}\right)^k$*, with* $k = 5$ *in both algorithms.*

following values of $d = \{1, 2, 3, 4, 5\}$. As with the previous data-set, we see that Algorithm 2 is always the best regardless of the dimensionality, and that the optimization procedure of Section 4.3 generally helps improve the results obtained with other algorithms. Once more, we can further illustrate these results with the help of the 2-dimensional projections of the data onto each of the subspaces found by the algorithms tested. These are in Appendix B.

6 Conclusions

This paper presents an algorithm to find that d-dimensional subspace (within the original feature space $\mathbb{R}^p$) where the Bayes error is minimized for a set of C homoscedastic Gaussian distributions. The algorithm ensures that the first dimension (feature) is that where the Bayes classification error is minimum. Consecutive dimensions are also guaranteed to minimize the Bayes error in the null space of the previous solution. The main result was presented in Theorem 1, and Eq. (4) defines the mechanism to find each of the 1-dimensional optimal solutions. This was extended to find the d-dimensional solution in Algorithm 1. We have also proposed a linear approximation of our method, which can be implemented as a low-cost approach summarized in Algorithm 3. Here, we showed that LDA is in fact similar in nature to our approximation, but is sometimes biased toward the incorrect classes. A set of criteria have been presented that can be used to determine where our approximation or LDA works. Finally, we have shown how our main result can be applied to improve upon those obtained by current feature extraction algorithms. Our experimental results demonstrate that such improvements are considerably large even when the assumption of homoscedasticity needs to be relaxed. Extensive experimental results on synthetic and real data have demonstrated the use of our algorithms in extracting low-dimensional feature representation. This is most noticeable when the subspace needs to be of less than five or four dimensions.

Acknowledgments

We thank the referees for their constructive comments. This research was partially supported by NIH under grant R01 DC 005241.

References

[1] B.B. Verbeck, M.V. Chafee, D.A. Crowe and A.P. Georgopoulos, "Parallel processing of serial movements in prefrontal cortex," Proceedings of the National Academy of Sciences of the USA, 99(20):13172-13177, 2002.

[2] G. Baudat and F. Anouar, "Generalized Discriminant Analysis Using a Kernel Approach," Neural Computation, 12(10):2835-2404, 2000.

[3] S.V. Beiden, M.A. Maloof and R.F. Wagner, "A general model for finite-sample effects in training and testing of competing classifiers," IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(12):1561-1569, 2003.

[4] R.A. Fisher, "The statistical utilization of multiple measurements," Annals of Eugenics, 8:376-386, 1938.

[5] K. Fukunaga and J.M. Mantock, "Nonparametric discriminant analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, 5:671-678, 1983.

[6] S. Geisser, "Discrimination, allocatory, and seperatory linear aspects," In *Classification and Clustering*, J. Van Ryzin, Ed., pp. 301-330, 1977.

[7] P.E. Gill, W. Murray and M.H. Wright, "Numerical Linear Algebra and Optimization, Vol. 1," Addison Wesley, 1991.

[8] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," *In Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[9] M. Loog, R. P. W. Duin and R. Haeb-Umbach,"Multiclass linear dimension reduction by weighted pairwise fisher criteria," IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(7):762-766, 2001.

[10] R. Lotlikar and R. Kothari, " Fractional-step dimensionality reduction," IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(6):623-627, 2000.

[11] J. Lu, K.N. Plataniotis and A.N. Venetsanopoulos, "Face recognition using LDA-based algorithms," IEEE Trans. Neural Networks, 14(1):195-200, 2003.

[12] A.M. Martinez and A.C. Kak, "PCA versus LDA," IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(2):228-233, 2001.

[13] A.M. Martinez and M. Zhu, "Where are linear feature extraction methods applicable?," IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(12):1934-1944, 2005.

[14] S. Michiels, S. Koscielny and C. Hill, "Prediction of cancer outcome with microarrays: a multiple random validation strategy," Lancet, 365(9458):488-492, 2005.

[15] S. Mika, G. Ratsch, J. Weston, B. Scholkopf and K. Muller, "Fisher Discriminant Analysis with Kernels," Proceedings of IEEE Neural Networks for Signal Processing Workshop, 1999.

[16] D.J. Newman, S. Hettich, C.L. Blake and C.J. Merz,"UCI Repository of machine learning databases," *http://www.ics.uci.edu/∼mlearn/MLRepository.html*, University of California, Irvine, Dept. of Information and Computer Sciences, 1998.

[17] P.J. Phillips, P.J. Flynn, T. Scruggs, K.W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the Face Recognition Grand Challenge," In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[18] C.R. Rao, "Linear statistical inference and its applications (second edition)," Wiley Interscience, 2002.

[19] M. Sommer, A. Olbrich and M. Arendasy, "Improvements in personnel selection with neural networks: A pilot study in the field of aviation psychology," International Journal of Aviation Psychology, 14(1):103-115, 2004.

[20] M. J. Schervish, "Linear discrimination for three known normal populations," Journal of Statistical Planning and Inference, 10:167-175, 1984.

[21] J. Yang, G.W. Xu, Q.F. Hong, H.M. Liebich, K. Lutz, R.M. Schmulling and H.G. Wahl, "Discrimination of type 2 diabetic patients from healthy controls by using metabonomics method based on their serum fatty acid profiles," Journal of Chromatography B–Analytical Tehcnologies in the Biomedical and Life Scienes, 813 (1-2): 53-58 DEC 25 2004.

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

Appendix A: Notation

In this paper, vectors are represented with bolded lower-case letters (e.g., $\mathbf{x}$), except for the mean feature vectors which (following classical notation) are represented as μ. Similarly matrices are represented using bolded capital letters (e.g., $\mathbf{S}_B$ for the between-class scatter matrix), except for the covariance matrix which is Σ. The vectors in the whitened space are represented with a hat symbol, e.g., $\hat{\mu}$. A set of vectors $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ can be represented from smallest to largest as $\{\mathbf{x}_{(1)}, \mathbf{x}_{(2)}, \ldots, \mathbf{x}_{(n)}\}$, where $\mathbf{x}_{(1)} \leq \mathbf{x}_{(2)} \leq \cdots \leq \mathbf{x}_{(n)}$ and $\mathbf{x}_{(i)} = \mathbf{x}_j$ for some (i, j) pair. Convex regions are described using calligraphic capital letters, e.g., $\mathcal{A}$. The orthogonal space to a vector $\mathbf{x}$ is described using the capitalized letter and an orthogonal symbol, $\mathbf{X}^{\perp}$. A detailed list of symbols and variables is as follows.

C	number of classes
d	dimensionality of the reduced space
$\mathbf{x}$	feature vector
$\mathbf{x}_{i,j}$	vector difference between $\mathbf{x}_i$ to $\mathbf{x}_j$
$\mathbf{X}^{\perp}$	feature space orthogonal to $\mathbf{x}$
μ	mean feature vector
Σ	covariance matrix
$N(\mu, \Sigma)$	Gaussian (Normal) distribution
$\mathbf{S}_B$	between-class scatter matrix
$\hat{\mu}$	whitened mean vector
$\hat{\mathbf{S}}_B$	whitened between-class scatter matrix
η_i	projected i^{th} mean vector
$\eta_{(i)}$	i^{th} smallest projected mean vector
$\mathcal{A}$	convex region
$g(\mathbf{v})$	Bayes error function in $\mathcal{A}$
$\mathcal{F}$	Inner product space
$\psi()$	vector mapping function
$\Psi()$	matrix mapping function

Appendix B: Experimental results (details and extensions)

In this section we extend on the tests reported in the paper and describe details of each of the experimental results reported in the main paper.

Our first experimental result on synthetic data shown in Fig. 1 includes 6 homoscedastic Gaussian distributions embedded in a 3-dimensional space with means located at: $(0, 0, 1)^T$, $(0, 0, -1)^T$, $(0, 4, 0)^T$, $(0, -4, 0)^T$, $(7, 0, 0)^T$, $(-7, 0, 0)^T$, respectively, and covariance matrices equal to one fourth of the identity matrix, i.e., $\Sigma = \mathbf{I}/4$. In our simulation, we randomly generated 500 samples from each of these distributions. The class distributions of these samples were shown in Fig. 1(a) and the most discriminant feature vector (dimension) obtained with LDA was shown in (b). We saw that, with LDA, almost all the classes overlap with one another, resulting in a 51% classification rate on this (training) set. The solution of the proposed algorithm was shown in Fig. 1(c), which results in a 91% classification rate. For comparison, we now show the results obtained with several of the other algorithms. These are in Fig. 8. In this figure we also show the 2-dimensional plots for each of the algorithms.

The results shown above corresponds to the use of synthetic homoscedastic data in $\mathbb{R}^3$. One could argue that discriminant analysis methods should be used in spaces of larger dimensionality. We now

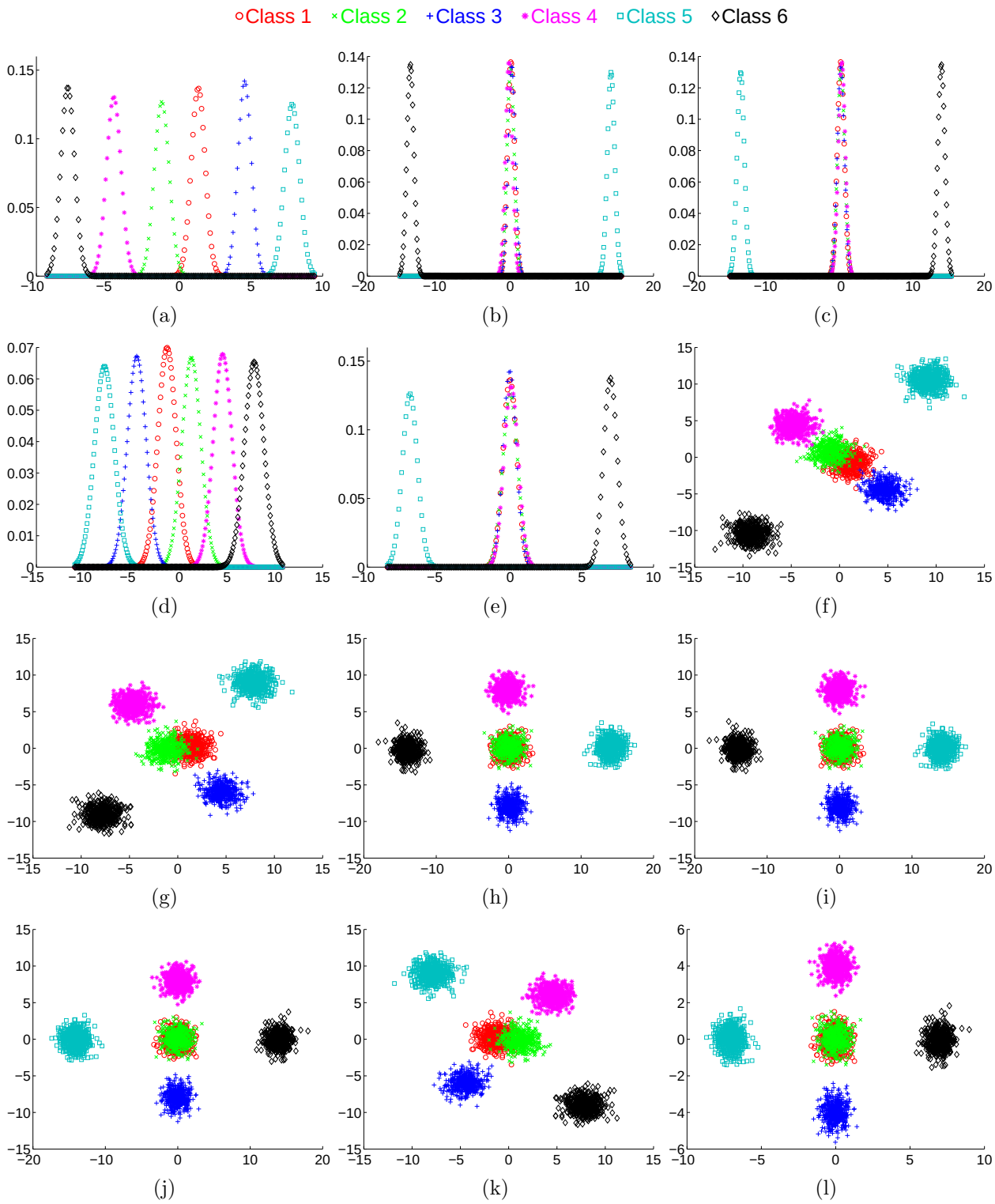


Figure 8: *Most discriminant dimension found by (a) FLDA, (b) aPAC,(c) Algorithm 3, (d) DFLDA, and (e) PCA-LDA. The 2-dimensional plots obtained are in (f) Algorithm 1, (g) FLDA, (h) aPAC, (i) LDA, (j) Algorithm 3, (k) DFLDA, (l) PCA-LDA, respectively.*

Dimensionality	Algorithm 1	LDA	FLDA	aPAC	Algorithm 3	DFLDA
1	82.56	48.72	72.6	52.08	58.44	72.4
2	96.52	86.4	91.64	88.72	95.2	91.12

Table 4: Recognition rates corresponding to each of the plots shown in Fig. 9.

Dimensionality	Algorithm 1	LDA	FLDA	aPAC	Algorithm 3	DFLDA
1	85.33	62.7	77.23	64.07	51.1	74.67
2	94.7	78.57	86.9	81.3	81.33	85.93

Table 5: Recognition rates corresponding to each of the plots shown in Fig. 10.

show a couple of examples drawn from $\mathbb{R}^{20}$. The first of these examples includes five homoscedastic (Gaussian distributed, with $\Sigma = \mathbf{I}$) pdf centered at $(0,\ldots,0)^T$, $(6,0,\ldots,0)^T$, $(0,12,0,\ldots,0)^T$, $(0,0,6,0,\ldots,0)^T$, and $(0,0,0,6,0,\ldots,0)^T$, respectively. Again, we randomly generated 500 samples from each of these distributions. The 1- and 2-dimensional results obtained with each of the algorithms are shown in Fig. 9. The corresponding classification rate (on this training data) are shown in Table 4. We see that the results obtained with Algorithm 1 are the best, followed by Algorithm 3 in the 2-dimensional cases and FLDA in the 1-dimensional representation.

Our final test using synthetic data includes six homoscedastic Gaussian distributions with $\Sigma = \mathbf{I}$. In this case, the class means are: $(0,\ldots,0)^T$, $(6,0,\ldots,0)^T$, $(0,6.6,0,\ldots,0)^T$, $(0,0,6.12,0,\ldots,0)^T$, $(0,0,6.12,12,0,\ldots,0)^T$, and $(0,0,6.12,0,12,0,\ldots,0)^T$. The 1- and 2-dimensional plots are in Fig. 10, and the corresponding classification rates in Table 5.

Another data-set used in our results was the image segmentation data of the UCI database. In Fig. 11 and Fig. 12 we show the 1D and 2D projections of the sample class distributions for each of the algorithms.

The other database from the UCI repository that we used was *LandSat*. The classification results were given in Table 3. The 2D plots (of the data projected onto the two most discriminant features found by each of the algorithms) are shown in Fig. 13.

The other data-set used in our experiments is the ETH-80 database described in detail in [6]. This database includes images of eight categories: apples, cars, cows, cups, dogs, horses, pears and tomatoes. Each of these categories is represented by the images of ten objects (e.g., ten cars) photographed from a total of 41 orientations. This means that we have a total of 410 images per category which gives a rich set of visual cues. In our experiments, we used the cropped images, where the objects have already been cropped out from the background, leaving the background pixels at zero-intensity. This is achieved by masking the cropped images with the object-masks provided by the authors of the database.

We also provide details for the results with the face recognition experiment on FRGC. Fig. 14 shows the cumulative match scores obtained for dimensionalities 5, 10 and 20.

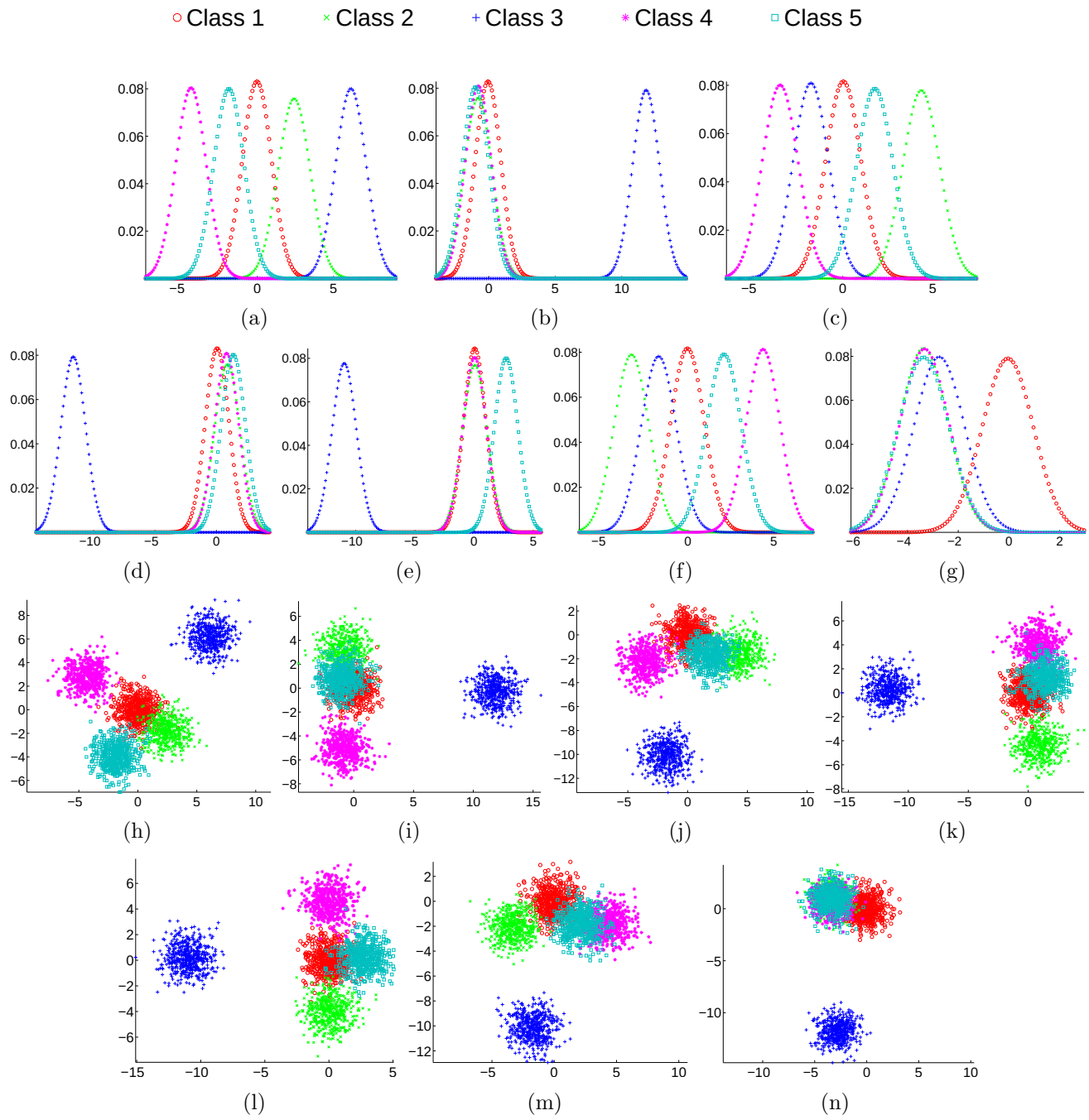


Figure 9: *Shown here is the first dimension obtained by: (a) Algorithm 1, (b) LDA, (c) FLDA, (d) aPAC, (e) Algorithm 3, (f) DFLDA, and (g) PCA-LDA. The 2-dimensional space are for: (h) Algorithm 1, (i) LDA, (j) FLDA, (k) aPAC, (l) Algorithm 3, (m) DFLDA, and (n)PCA-LDA.*

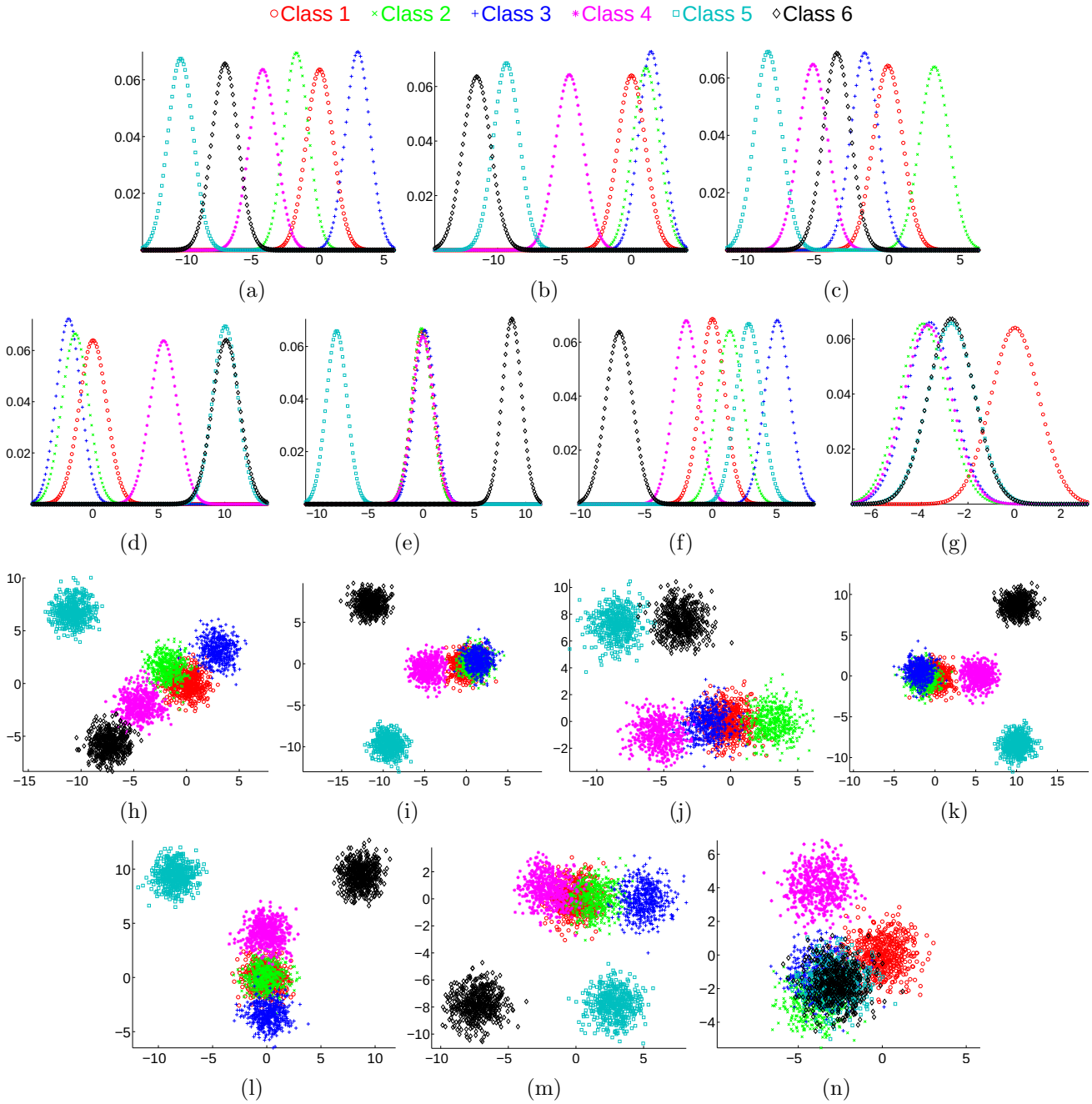


Figure 10: *Shown here is the first dimension obtained by: (a) Algorithm 1, (b) LDA, (c) FLDA, (d) aPAC, (e) Algorithm 3, (f) DFLDA, and (g) PCA-LDA . The 2-dimensional space are for: (h) Algorithm 1, (i) LDA, (j) FLDA, (k) aPAC, (l) Algorithm 3, (m) DFLDA, (n) PCA-LDA.*

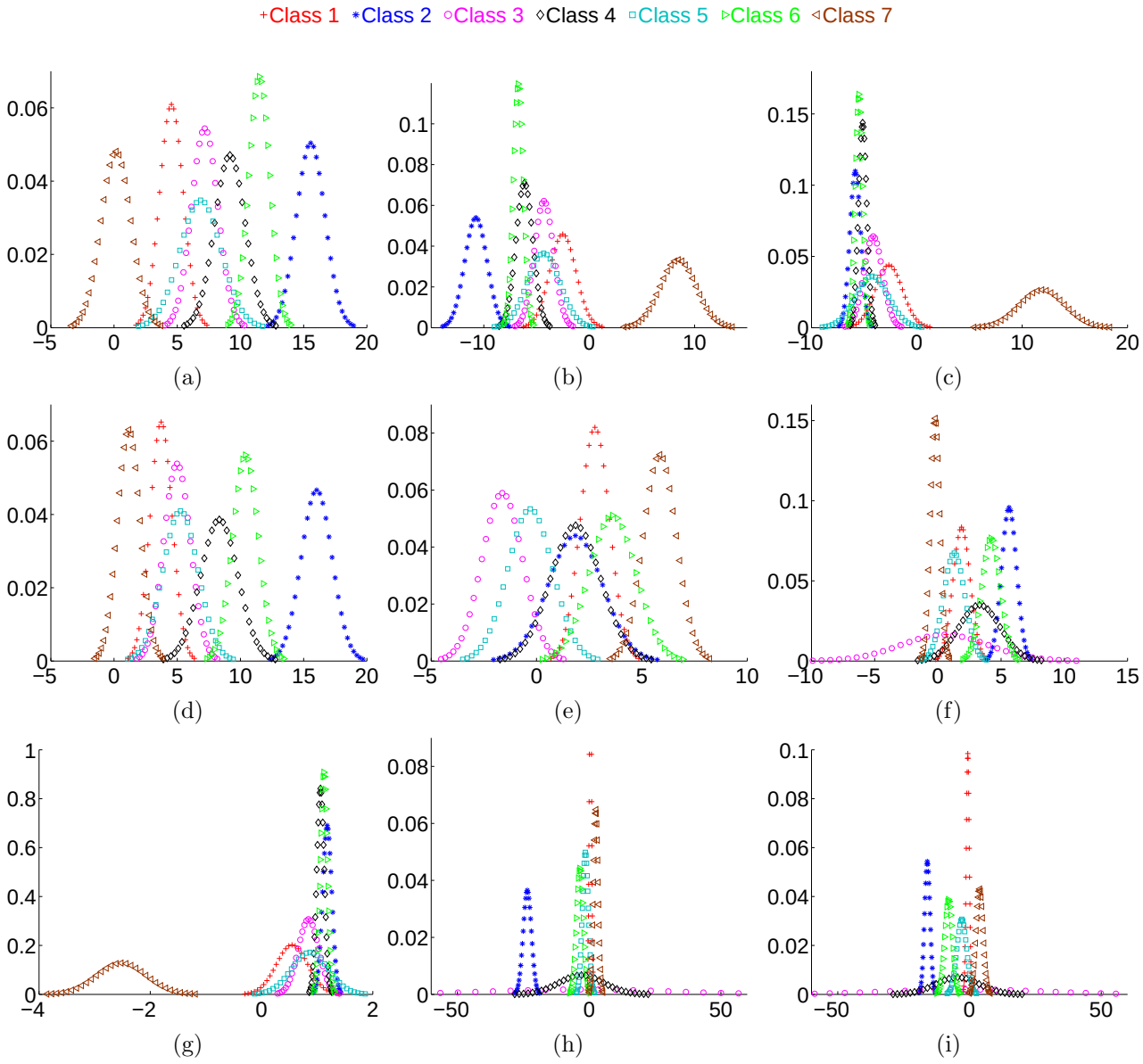


Figure 11: *One-dimensional plots of the seven class distributions of the UCI image segmentation set. Shown here are the projected results obtained by: (a) Algorithm 1, (b) Algorithm 3, (c) LDA, (d) aPAC, (e) FLDA, (f)DFLDA, (g) PCA-LDA, (h) KLDA, and (i) Algorithm 2.*

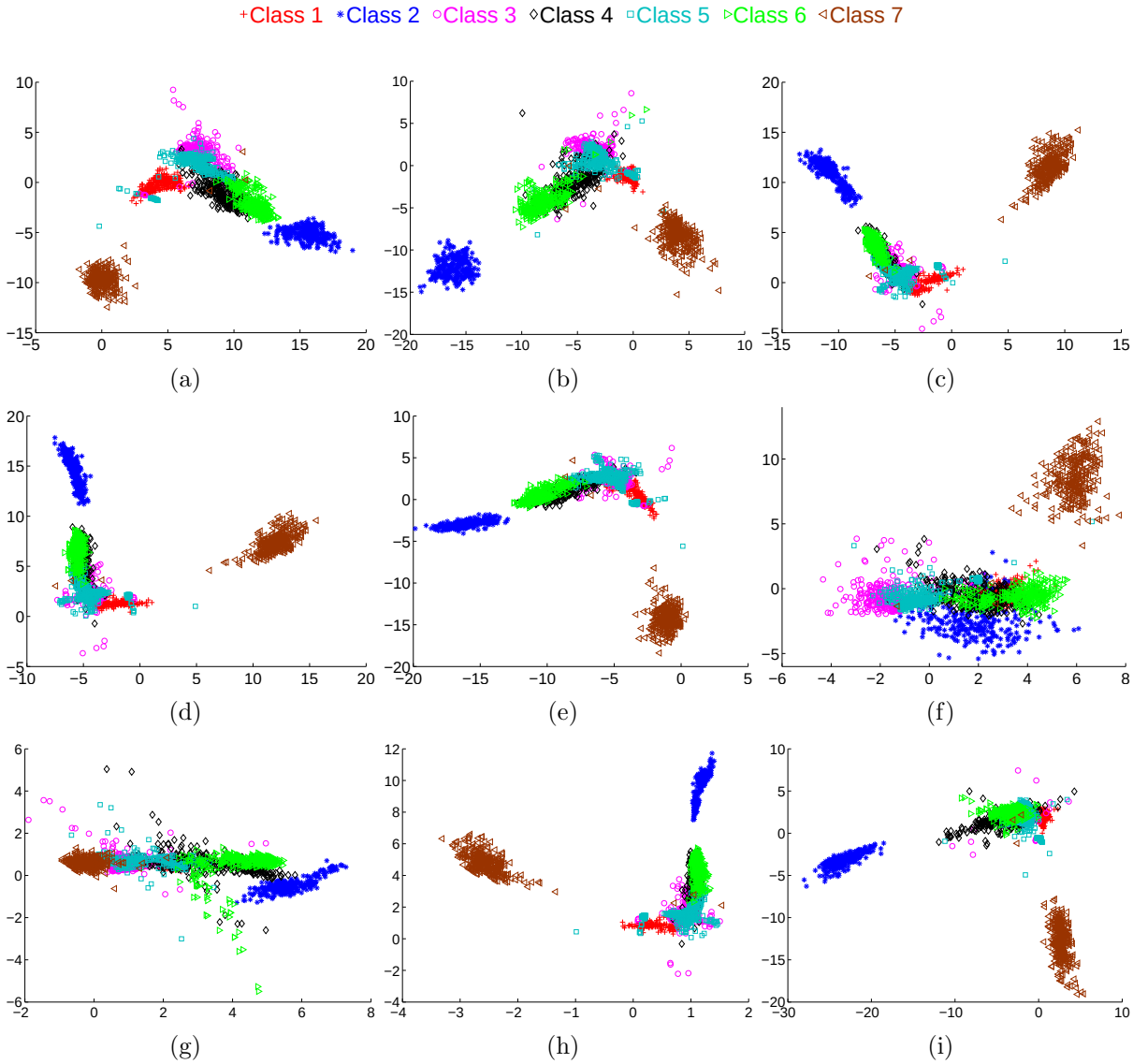


Figure 12: *Shown here are the 2-dimensional projections obtained with each of the algorithms tested: (a) Algorithm 1, (b) Algorithm 2,(c) Algorithm 3, (d) LDA, (e) aPAC, (f) FLDA, (g) DFLDA, (h) PCA-LDA and (i) KLDA.*

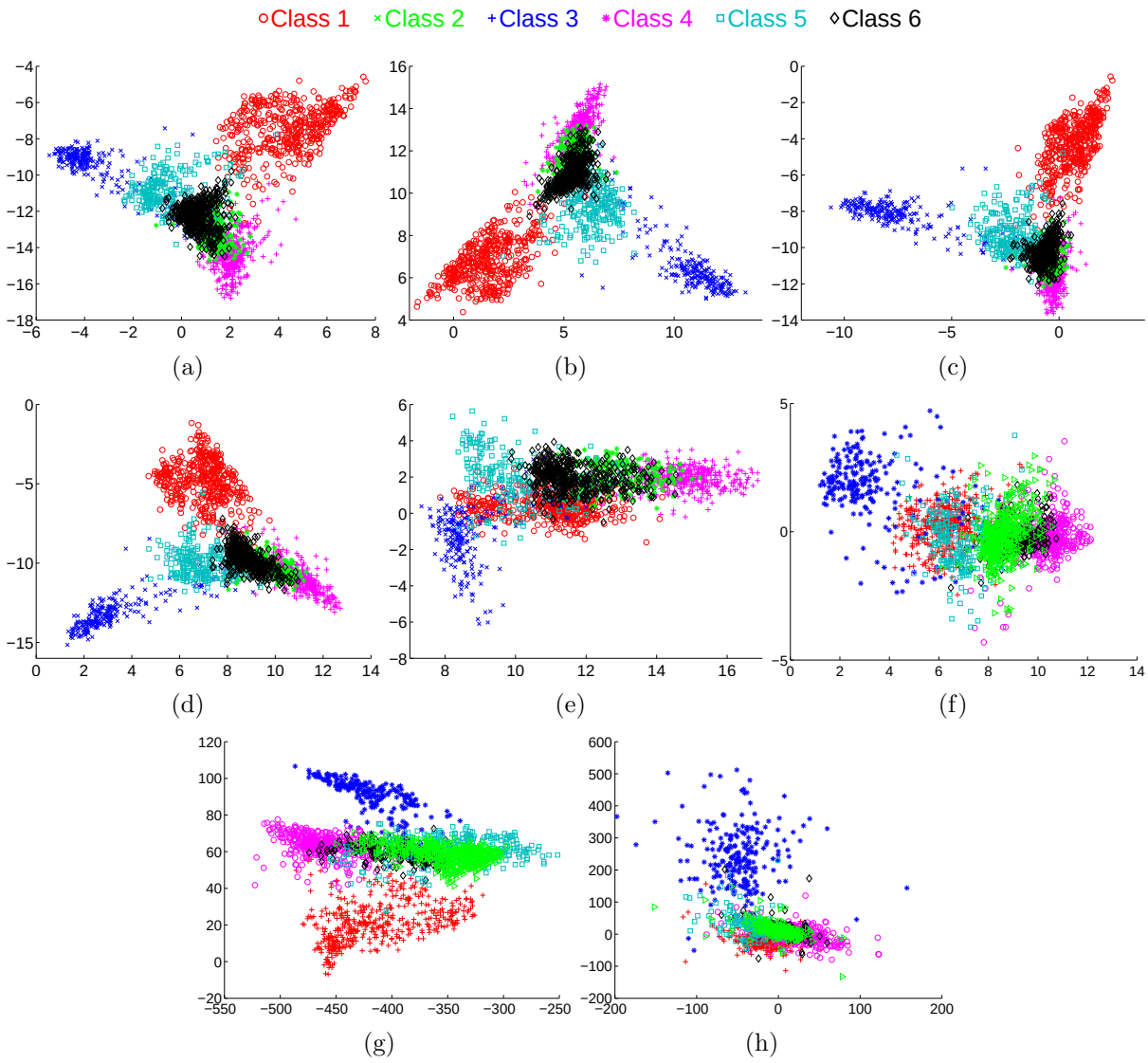


Figure 13: *Projections of the LandSat data onto the two most discriminant feature vectors found by each of the algorithms. Results found by: (a) Algorithm 1, (b) Algorithm 3, (c) LDA, (d) aPAC, (e) FLDA, (f) DFLDA, (g) PCA-LDA, and (h) Algorithm 2.*

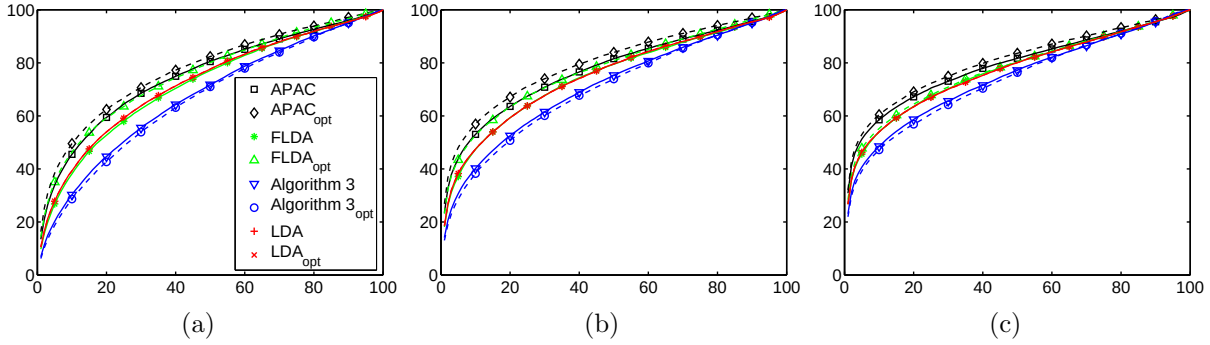


Figure 14: *Cumulative match scores obtained with FRGC experiment for dimensionalities (a)* $d = 5$*, (b)* $d = 10$*, and (c)* $d = 20$*.*

Appendix C: Practical Issues

The optimization algorithms proposed to find the Bayes optimal dimension can be run faster by realizing that the whitened class means of C classes lie on a $C-1$ dimensional subspace.

In Algorithm 1 this can be done by projecting the p-dimensional whitened class means $\hat{\mu}_i$ onto the range space of the whitened between-class scatter matrix. That is, $\check{\mu}_i = \widehat{\mathbf{V}}_B^T \hat{\mu}_i$, where $\hat{S}_B \widehat{\mathbf{V}}_B = \widehat{\mathbf{V}}_B \Lambda_B$. If the solution obtained by running Algorithm 1 with $\check{\mu}_i$ is $\check{\mathbf{V}}_{BAYES}$, the linear projection vector from p to d dimensions will be

$$\mathbf{V}_{BAYES} = \bar{\mathbf{V}} \bar{\Lambda}^{-1/2} \widehat{\mathbf{V}}_B \check{\mathbf{V}}_{BAYES},$$

where $\bar{\mathbf{V}}$ and $\bar{\Lambda}$ are the eigenvector and eigenvalue matrices of $\bar{\Sigma}$. The same trick can be used to speed up Algorithm 3.

Furthermore, to obtain the kernel d-dimensional subspace, one can reduce the computational complexity by projecting the whitened mean locations in the kernel space $\widehat{\mu}_i^{\psi}$ to a $C-1$ dimensional subspace. Let $\hat{\mathbf{S}}_B^{\psi} = \sum_{i=1}^{C} (\hat{\mu}_i^{\psi} - \hat{\mu}^{\psi})(\hat{\mu}_i^{\psi} - \hat{\mu}^{\psi})^T$, where $\hat{\mu}^{\psi}$ is the average over all $\hat{\mu}_i^{\psi}$. Then, Algorithm 2 will run faster if one uses $\check{\mu}_i^{\psi} = \hat{\mathbf{V}}_B^{\psi T} \hat{\mu}_i^{\psi}$, with $\hat{\mathbf{S}}_B^{\psi} \hat{\mathbf{V}}_B^{\psi} = \hat{\mathbf{V}}_B^{\psi} \hat{\Lambda}_B^{\psi}$. Assuming that the solution obtained by Algorithm 2 is $\check{\mathbf{W}}_{BAYES}$, the final projection matrix is given by, $\mathbf{V}_K = \Psi(\mathbf{X}) \Gamma \Lambda^{\Psi^{-1/2}} \hat{\mathbf{V}}_B^{\psi} \check{\mathbf{W}}_{BAYES}$.