Sparse Kernels for Bayes Optimal Discriminant Analysis

Onur C. Hamsici and Aleix M. Martinez
Department of Electrical and Computer Engineering
Ohio State University, Columbus, OH 43210
{hamsicio,aleix}@ece.osu.edu

Abstract

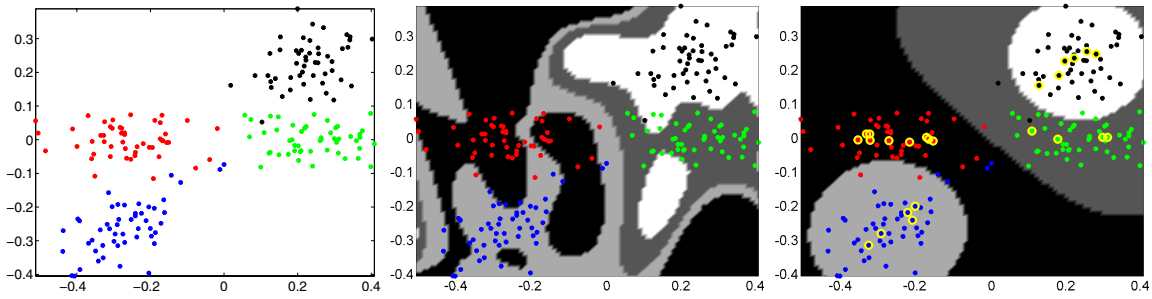
Discriminant Analysis (DA) methods have demonstrated their utility in countless applications in computer vision and other areas of research – especially in the C class classification problem. The most popular approach is Linear DA (LDA), which provides the $C-1$-dimensional Bayes optimal solution, but only when all the class covariance matrices are identical. This is rarely the case in practice. To alleviate this restriction, Kernel LDA (KLDA) has been proposed. In this approach, we first (intrinsically) map the original non-linear problem to a linear one and then use LDA to find the $C-1$-dimensional Bayes optimal subspace. However, the use of KLDA is hampered by its computational cost, given by the number of training samples available and by the limitedness of LDA in providing a $C-1$-dimensional solution space. In this paper, we first extend the definition of LDA to provide subspace of $q < C-1$ dimensions where the Bayes error is minimized. Then, to reduce the computational burden of the derived solution, we define a sparse kernel representation, which is able to automatically select the most appropriate sample feature vectors that represent the kernel. We demonstrate the superiority of the proposed approach on several standard datasets. Comparisons are drawn with a large number of known DA algorithms.

1. Introduction

A major task in pattern recognition and machine learning is to find that representation where a set of C classes is best discriminated. For example, in computer vision, we may want to find that image representation that allows us to discriminate between a set of C object categories. In this application, we first extract a set of p features from the image – be it geometric or appearance features or a combination of them. Then, we want to use an algorithm that tells us which q linear or non-linear combinations of the original features best discriminate between the samples of different classes. This will reduce the original representation from p to q features (dimensions). The rest of the features $(p-q)$ are eliminated because they contain information irrelevant to the classification of the classes and would make the identification of novel (previously unseen) samples very difficult – they could even make a test sample closer to the incorrect class [2]. *Therefore, ideally, we want to make q as small as possible*. This will guarantee we only use the discriminant features while eliminate the distracters (i.e., noisy features).

Discriminant Analysis (DA) is a set of algorithms commonly used for this task. In DA, we assume the samples are drawn from a set of probability density functions (pdf). These are generally assumed to be Normal or a mixture of these [16]. Probably the most known and used algorithm is Linear Discriminant Analysis (LDA) [1, 2] and extensions [16]. The appeal of LDA comes from the fact that this provides the Bayes optimal $C-1$-dimensional subspace solution whenever the class pdf share a common covariance matrix. By Bayes optimal, it is understood that we can reduce the original representation from p dimensions to $C-1$ with no classification loss (i.e., maintaining the same Bayes classification error). Hence, all the features we eliminate are redundant (noise) and would only serve as distracters in the classifications of subsequent test feature vectors.

Unfortunately, LDA is limited to the case of equal class pdf, which is rare the case in practice. A simple (illustrative) example is shown in Fig. 1(a), where we have four Normal distributions with distinct covariances. In this example, the Bayes optimal classifier is non-linear. To resolve this issue, a method called Kernel LDA (KLDA) has been proposed [9]. In KLDA, we first use a kernel matrix to convert a non-linearly separable problem into a linearly separable one. This can be achieved by defining a symmetric, positive-definite matrix $\mathbf{K}$ (known as the Gram matrix) which is the metric defined by the sample feature vectors. In other words, this metric defines the non-linearity of the sample vectors. By undoing this non-linearity, we can find the Bayes optimal $C-1$-dimensional subspace as in LDA. Then, the non-linear solution is obtained by mapping the LDA solution back into the original space. This idea is very useful because kernels can be defined using a dot product in a Hilbert space (i.e., a reproducing kernel Hilbert space)



(a) (b) (c)

Figure 1. *(a) Shown here are samples drawn from four heteroscedastic Gaussian distributions in $\mathbb{R}^2$. (b) Shows the classification results (regions with different constrasts) obtained in the 1-dimensional Kernel LDA space. (c) Shows the classification results in the nonlinear Bayes optimal 1-dimensional space obtained in the sparse kernel feature space represented by the samples marked with yellow blobs.*

[15].

Unfortunately, when the dimensionality of the solution in the kernel space is less than $(C-1)$ this method may fail. Fig. 1(b) demonstrates how KLDA fails when only the most discriminant dimension is extracted.

Moreover, the use of a kernel as shown above, is impaired by the computational cost of the resulting algorithm. Note that since the Gram matrix is defined using dot products of all training samples, the mapping from the original to the kernel space is computationally demanding. Also, the requirement on memory sometimes surpasses those available. This is especially true whenever the number of classes C is large. This issue has been previously addressed in Support Vector Machines (SVM) [12] and in Kernel Principal Components Analysis (KPCA) [13]. However, the algorithms defined to sparsify Kernel Fisher Discriminant Analysis (KLDA) are not useful in practice [10]. This is because the sparse versions of KLDA is defined as an optimization routine, where the l_1 (linear sparse KLDA, called SKDA$_1$) or l_2 (called SKDA$_2$) norm of the classification error is minimized with an l_1 norm penalty on the solution vector. This optimization procedure is defined in an M^2+1 dimensional space for M training samples. While we need an algorithm to solve the complexity problem for large M, the SKDA$_1$ or SKDA$_2$ requires an optimization algorithm that has complexity proportional to M^2 . Hence, the usage of this algorithm is limited to either small datasets, or high-end, powerful computers.

Therefore, we need another approach to solve the complexity problem in DA. To address this problem of the kernel approach, we will use a sparse matrix representation of $\mathbf{K}$. We have already stated that $\mathbf{K}$ describes the metric defining the non-linear classification of the classes. Hence, we can simplify our problem to determining that subset of sample feature vectors that suffice to approximate this metric. In other words, the kernel matrix will be given by a small subset of the original samples – what is known as *sparse* matrix.

As stated above another *key* question in DA is to find that q-dimensional subspace which provides the Bayes optimal projection – in the sense that there is no classification loss when reducing the feature space from p to q dimensions. LDA and, analogously, KLDA provide the Bayes optimal projection for $q = C-1$. An important open question in DA is how to find subspaces with $q < C-1$ where the Bayes error is minimized. In general, this problem does not have a solution, because the Bayes error cannot be calculated in spaces of 2 or more dimensions (since this involved nonlinear integration with no known solution). Nonetheless, the problem is solvable for $q = 1$. As shown in [3], the Bayes error in the one dimensional space is given by

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

where Φ is the cumulative distribution function (cdf) of the 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

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

is the mean feature vector μ_i of the i^{th} class after whitening and projection onto the one-dimensional subspace defined by the unit vector $\mathbf{v}$.

Hence, in principle, it should be possible to use a kernel to map the data into a space where this adapts to the linear constraint and there use (1) to find that one-dimensional subspace where the Bayes error is minimized. We summarize the work in [5] to show how this can be done in Section 2. Such an approach still requires that we optimize the kernel function to correctly map the data to a space where we can use Eq. (1). To make this step tractable, we introduce a sparse matrix representation that automatically selects the most representative samples of the kernel metric, allowing us to obtain the result shown in Fig. 1(c). Derivations and

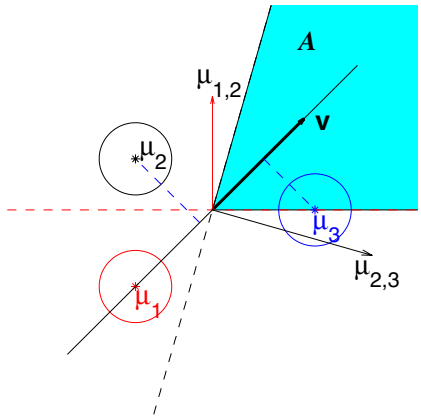


Figure 2. *This figure shows a basis* $\mathbf{v}$ *where the sequence of projected, whitened class means is* $\eta_1 \leq \eta_2 \leq \eta_3$. *Now, note that the color filled region defines the area where all the possible bases* $\mathbf{v}$ *that produce the same sequence are. More accurately, this 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* $\mu_{i,j} = \mu_j - \mu_i$.

details of this approach are in Sections 3-5. We provide extensive experimental results in Section 6. Conclusions are in Section 7.

2. The Bayes Optimal One-dimensional Subspace

In this section, we provide the derivations for obtaining the Bayes optimal one-dimensional subspace for the case of equal class covariances. Following (1), we assume we have whitened the class pdf, i.e., $\widehat{\Sigma} = \mathbf{I}$, and that we have equal class priors. Now, we note 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 colored region represents the set of all one-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 [5]. This allows to state the following result.

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 *Normal distributions with identical class covariance matrix. Let* $g(\mathbf{v})$ *be the Bayes error function of the* C *Normal distributions in* $\mathcal{A}$. *Then, the Bayes error function* $g(\mathbf{v})$ *for all* $\mathbf{v} \in \mathcal{A}$ *is convex.*

Proof (sketch)[5]. It is noticed that the Hessian of the Bayes error function is positive semi-definite in $\mathcal{A}$, which means that

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

is convex in $\mathcal{A}$. □

The 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 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. We also need to restrict our search space with those vectors that have a unit length only, i.e., $\mathbf{v}^T\mathbf{v} = 1$. Since the Bayes error function is monotone decreasing i.e., $g(\alpha\mathbf{v}) > g(\mathbf{v})$ for $\alpha < 1$, we can relax the non-convex condition to a convex one, i.e., $\mathbf{v}^T\mathbf{v} \leq 1$. Hence, the new search space $\mathcal{B} = \mathcal{A} \bigcap \mathbb{B}^p$, where $\mathbb{B}^p$ is the p-dimensional unit ball, which is convex.

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. Finally, the optimal solution $\mathbf{v}$ to our problem is given by

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

From now on we call this algorithm as Bayes-optimal Linear Discriminant Analysis (BLDA). Now we extend this linear feature extraction method to nonlinear by using the kernel trick.

3. Kernel-Bayes solution

In practice it is necessary to relax the assumption of equal covariances. We can achieve this by means of the well-known kernel trick, which permits to perform the calculations in the original space. As in LDA, the criterion of KLDA will aim at the least squares solution maximizing the distance between the projected class means. Thus, this solution will be biased toward those class means that are farthest apart. By using our result provided in Theorem 1, one can resolve this problem.

Let μ_i^ψ represent the mean of class i in a high dimensional space 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

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

This equation can be rewritten as follows

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

where $\Psi(\mathbf{X}^i) = \left(\psi(\mathbf{x}_1^i),\ldots,\psi(\mathbf{x}_{n_i}^i)\right)$ is the matrix of sample vectors in $\mathcal{F}$, $\mathbf{I}$ is the identity matrix and $\mathbf{1}_{n_i}$ is a $n_i \times n_i$ 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 $\Psi(\mathbf{X}) = (\psi(\mathbf{x}_1),\ldots,\psi(\mathbf{x}_l))$. By 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

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

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

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

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

$$\mathbf{V}_K = \Psi(\mathbf{X})\Gamma{\Lambda^{\Psi}}^{-1/2}\mathbf{w}.$$

From now on this approach will be called Kernel Bayes-optimal Linear Discriminant Analysis, or KBLDA for short.

4. Sparse Representation

Thus far, we have been able to define a kernel-based DA algorithm that can find the Bayes optimal one-dimensional subspace representation to discriminate among C classes. As stated in the introduction of this paper, however, an important disadvantage of this nonlinear (kernel) extension is its computational inefficiency. This is because we use all the samples to represent the kernel space. Unfortunately, the algorithms defined to deal with this problem are generally more complex than the original problem [10]. Furthermore, this method requires to define the penalty coefficient which determines the sparseness of the data representation, which is known to be impractical. Ideally, we want an algorithm that can determine the most appropriate sparseness automatically and without the use of computationally expensive optimization methods, such as the leave-one-out test. One algorithm that accomplishes this is presented in [13] for the case of PCA. This method removes those samples that are redundant in describing the kernel (metric) under use. The maximum likelihood estimate (mle) of the sample weights are calculated under an underlying probabilistic model with a noise term. In this section, we follow this probabilistic approach to eliminate those samples that are *redundant* to represent the metric given by our kernel Bayes optimal solution.

The main assumption underlying our model is that the data covariance matrix $\mathbf{S}$ includes white noise (i.e., the redundancy is assumed to be noise). This means, we can write

$$\mathbf{S} = \sigma^2\mathbf{I} + \sum_{i=1}^{M}u_i\psi(x_i)\psi(x_i)^T = \sigma^2\mathbf{I} + \Psi(\mathbf{X})\mathbf{U}\Psi(\mathbf{X})^T,$$

where u_i is the weight associated with sample i, $\mathbf{U}$ is a diagonal matrix with the i^{th} diagonal element equal to u_i, σ^2 is the noise variance, and M is the number of samples. It is known that when σ^2 is zero, the mle of $u_i = 1/M$. In general, when there is a specified amount of noise, most of the weight parameters u_i that are associated with the redundant (or noisy) samples will be zero. Therefore, we can select those samples that are necessary to represent the data variance.

To estimate the mle of u_i we use the fast estimation algorithm proposed in [14]. If we assume that all of the samples are iid, the log-likelihood of the weight parameters are defined as

$$\mathcal{L}(u) = -\frac{1}{2}\left(M\log|\mathbf{S}| + tr(\Psi(\mathbf{X})^T\mathbf{S}^{-1}\Psi(\mathbf{X}))\right).$$

Calculating $|\mathbf{S}|$ and $tr(\Psi(\mathbf{X})^T\mathbf{S}^{-1}\Psi(\mathbf{X}))$ for infinite dimensionality may be problematic. Fortunately, we can use some tricks to address these problems. First, we rewrite $\mathbf{S}$ as

$$\mathbf{S} = \mathbf{S}_{-i} + u_i\psi_i\psi_i^T,$$

where $\mathbf{S}_{-i} = \sigma^2\mathbf{I} + \sum_{k\neq i}u_k\psi_k\psi_k^T$ is the covariance matrix obtained with all the terms but the i^{th} term, and we have used $\psi_i = \psi(x_i)$ for brevity. Hence,

$$\begin{aligned}|\mathbf{S}| &= |\mathbf{S}_{-i}||1 + u_i\psi_i^T\mathbf{S}_{-i}^{-1}\psi_i|, \\ \mathbf{S}^{-1} &= \mathbf{S}_{-i}^{-1} - \frac{\mathbf{S}_{-i}^{-1}\psi_i\psi_i^T\mathbf{S}_{-i}^{-1}}{u_i^{-1} + \psi_i^T\mathbf{S}_{-i}^{-1}\psi_i},\end{aligned}$$

where for the second equality we use the Woodbury matrix inversion [4]. Thus, the log-likelihood function can be restated as,

$$\begin{aligned}\mathcal{L}(u) &= -M/2\log|\mathbf{S}_{-i}| - tr(\Psi(\mathbf{X})^T\mathbf{S}_{-i}^{-1}\Psi(\mathbf{X}))/2 \\ &- M/2\log|1+u_i\psi_i^T\mathbf{S}_{-i}^{-1}\psi_i| \\ &+ \frac{tr(\Psi(\mathbf{X})^T\mathbf{S}_{-i}^{-1}\psi_i\psi_i^T\mathbf{S}_{-i}^{-1}\Psi(\mathbf{X}))}{u_i^{-1}+\psi_i^T\mathbf{S}_{-i}^{-1}\psi_i}/2.\end{aligned}$$

Now, let $q_i = \Psi(\mathbf{X})^T\mathbf{S}_{-i}^{-1}\psi_i, \quad s_i = \psi_i^T\mathbf{S}_{-i}^{-1}\psi_i$. Since $\mathcal{L} = \mathcal{L}(u_{-i}) + l(u_i)$, we have

$$\frac{\partial\mathcal{L}}{\partial u_i} = \frac{\partial l(u_i)}{\partial u_i} = -\frac{Ms_i(1+u_is_i) - q_i^2}{2(1+u_is_i)^2}.$$

As shown in [14], $\mathcal{L}$ has a unique maximum at $u_i = (q_i^2 - Ms_i)/(Ms_i^2)$, if $q_i^2 > Ms_i$, and at $u_i = 0$ if $q_i^2 \leq Ms_i$. This means that for a given subset of the samples, we can find that u_i maximizing the likelihood. However, once we modify the weight of the sample ψ_i, $\mathbf{S}$ will change. Thus, we need to update its value. This suggests an iterative algorithm where we add, delete or re-estimate the model parameters iteratively, leading to the mle of u.

By using this iterative procedure we can find those samples that are not related with the noise – the relevant ones. This is given by the samples that have nonzero weights. Let $\tilde{\Psi} = \left(\psi_1^*u_1^*, \ldots, \psi_{\tilde{M}}^*u_{\tilde{M}}^*\right)$ define the relevant data that is independent of noise, with ψ_i^* representing the samples with a nonzero weight u_i^*. Then the eigenvectors of the average within class scatter matrix represented by these samples are given by $\tilde{\mathbf{W}}^\psi = \tilde{\Psi}\tilde{\Gamma}$. Hence, $\tilde{\mathbf{N}} = \frac{1}{C}\sum_{i=1}^{C}\tilde{\mathbf{K}}_i(\mathbf{I} - \mathbf{1}_{n_i})\tilde{\mathbf{K}}_i^T$, with $\tilde{\mathbf{K}}_i = \tilde{\Psi}(\mathbf{X})^T\Psi(\mathbf{X}^i)$, which allows us to rewrite the spectral decomposition as

$$\tilde{\Gamma}^T\tilde{\mathbf{N}}\tilde{\Gamma} = \tilde{\Lambda}^\Psi.$$

Note that here $\tilde{\mathbf{N}}$ is only $\tilde{M}\times\tilde{M}$ where $\tilde{M} << M$. Hence, our final solution is given by

$$\tilde{\mathbf{V}}_K = \tilde{\Psi}(\mathbf{X})\tilde{\Gamma}\tilde{\Lambda}^{\Psi^{-1/2}}\tilde{\mathbf{w}},$$

where $\tilde{\mathbf{w}}$ is calculated as above but, now, from the computed sparse representation. The new one-dimensional solution vector $\tilde{\mathbf{V}}_k$ is represented with the relevant $\tilde{M}$ samples. From now on we call this algorithm the Sparse Kernel Bayes-optimal Linear Discriminant Analysis, SKBLDA.

5. The d-dimensional Subspace

To find a subspace solution of more than one dimension, we recursively apply our algorithms (BLDA, KBLDA and SKBLDA) to the null space of the previously obtained subspace. After applying an algorithm, one obtains the first subspace solution, $\mathbf{v}_1$. We then project the class means to the null-space of $\mathbf{v}_1$, where we re-apply the same algorithm to obtain $\mathbf{v}_2$. More generally, our algorithms can be recursively applied to find that d-dimensional solution from any p-dimensional space, with $d \leq \min(p, C-1)$.

This is summarized in Algorithm 1.

Algorithm 1 (d-dimensional subspace).

Let $\hat{\mu}_i$ be the whitened mean locations.
Calculate $\mathbf{v}_1$ using BLDA, KBLDA or SKBLDA and the mean locations $\hat{\mu}_i$. Set $\mathbf{V}_1 = \mathbf{v}_1$.
for *$i = 2$ to d* ***do***
  Project $\hat{\mu}_i$ to the null space $\mathbf{V}_{i-1}^\perp$ of the current solution $\mathbf{V}_{i-1}$, i.e., $\tilde{\mu}_i = \mathbf{V}_{i-1}^{\perp^T}\hat{\mu}_i$.
  Calculate $\mathbf{v}_i$ using BLDA, KBLDA or SKBLDA with $\tilde{\mu}_i$.
  $\mathbf{V}_i = \left(\mathbf{V}_{i-1}, \mathbf{V}_{i-1}^\perp\mathbf{v}_i\right)$.
end for

6. Experimental Results

To illustrate the use of BLDA, KBLDA, SKBLDA, we compare them to LDA, KPCA, SKPCA (Sparse KPCA), KLDA , SKLDA (Sparse KLDA), and Fractional LDA [7] (FLDA, which is a method designed to find lower dimensional representations within LDA's solution). In our comparative studies, we will also use an implementation of the PCA-LDA algorithm, where PCA is first used to reduce the dimensionality of the original space. 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.

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 [6]. 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 [8].

A common feature space in object recognition is one that uses the derivatives of the Gaussian filter. In particular, we considered the use of the magnitude of the gradient and the Laplacian operator, which generate rotation-invariant information. The gradient and Laplacian are computed using three scales, i.e., $\sigma = \{1, 2, 4\}$. 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. This image representation is 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 shows the average classification rates (in percentages).

The importance of the sparse representation comes from the number of samples that are used to represent the kernel, which directly effects the time complexity. In KLDA and KBLDA only 196 samples are necessary to successfully represent our metric (out of a total of $3,239$ training samples). SKBLDA does lose some of the recognition capacity. This is the payoff to be considered.

Image segmentation: Our next test uses the image segmentation data-set of the UCI machine learning repository [11]. 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.[1] These represent: the pixel location of the patch 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(a) summarizes the successful classification rate on the testing set for the following values of the dimensionality $d = \{1, 2, 3, 4, 5, 6\}$. The sparse extensions of KLDA and KPCA reduce the complexity of the algorithms without any significant loose in the recognition rate. Our sparse algorithm SKBLDA selected 25 samples from the original 210 training samples.

In this case, we were also able to test the regularized KDA, SKDA$_1$ and SKDA$_2$. With the penalty coefficient optimized to $c = .001$ using 10-fold cross validation, the best recognition rates obtained with regularized KDA, SKDA$_1$ and SKDA$_2$ were 86.6% (rbf kernel $\sigma = .4$), 23.29% (rbf kernel $\sigma = .4$) and 33.86% (polynomial kernel $k = 1$). Although using C classifiers in a one-versus all scheme, this corresponds to doing the classification in C-1 dimensions. These algorithms could not perform better than the proposed BLDA, KBLDA and SKBLDA.

Satellite imagery: Our final test uses the Landsat data of the Statlog project defined in [11]. 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 2(b) summarizes the classification rates obtained with each of the algorithms tested for the following values of $d = \{1, 2, 3, 4, 5\}$. SKBLDA performs as good as its non-sparse version with only 236 samples out of the $4,435$ initial training samples.

[1] 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.

7. Conclusions

In this paper we have proposed an algorithm to extract a single dimensional space such that the Bayes error is minimized for the C class classification problem. This includes a kernel extension of the algorithm. A major difficulty in the computational complexity of the nonlinear extension is eliminated by using a fast sparse feature extraction algorithm prior to our solution. Finally, both kernel and linear methods are extended to d-dimensional spaces where each dimension is constrained to be orthogonal to the others. The advantages of our theoretical results are supported by the experiments obtained from 3 different datasets, where we showed that our approach performs the best.

Acknowledgments

This research was partially supported by NIH under grant R01 DC 005241.

References

[1] R.A. Fisher, "The statistical utilization of multiple measurements," Annals of Eugenics, 8:376-386, 1938. 1

[2] K. Fukunaga, "Introduction to Statistical Pattern Recognition," Academic Press, 1990. 1

[3] S. Geisser, "Discrimination, allocatory, and seperatory linear aspects," In *Classification and Clustering*, J. Van Ryzin, Ed., pp. 301-330, 1977. 2

[4] G. Golub and C. Van Loan, "Matrix Computations," The Johns Hopkins University Press, 1996. 5

[5] O.C. Hamsici and A.M. Martinez, "Bayes Optimality in Linear Discriminant Analysis," Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007. 2, 3

[6] 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. 5

[7] R. Lotlikar and R. Kothari, " Fractional-step dimensionality reduction," IEEE Transactions on Pattern Analysis and Machine Intelligence, 22(6):623-627, 2000. 5

[8] 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. 5

Dimensionality (d)	1	2	3	4	5	6	7
BLDA	57.53	67.04	72.23	72.80	74.02	76.62	78.51
KBLDA	**84.21**	**86.43**	**88.75**	**89.05**	**91.01**	**91.86**	**92.47**
SKBLDA	72.56	81.71	82.74	84.97	85.46	87.90	90.30
LDA	46.89	62.35	70.24	70.73	74.51	76.98	78.50
FLDA	50.09	60.67	67.74	72.04	75.43	77.38	78.50
PCA-LDA	34.18	56.34	61.65	64.60	64.51	62.68	67.10
KLDA	63.20	69.51	73.81	80.18	81.10	88.14	**92.44**
SKLDA	57.50	68.96	74.60	78.17	78.05	87.80	90.30
KPCA (61.16)	24.88	36.71	44.39	48.81	49.63	49.73	52.35
SKPCA (61.19)	33.26	36.46	44.33	48.78	49.57	49.97	52.41

Table 1. Average classification rates in the object categorization task. The kernel function used was $\psi(\mathbf{x})^T\psi(\mathbf{y}) = \left(\mathbf{x}^T\mathbf{y}\right)^k$. The parameter k was optimized (independently in each algorithm) using the leave-one-out test.

Dimensionality (d)	1	2	3	4	5	6
BLDA	**71.24**	**80.86**	85.24	**88.05**	**89.95**	**90.47**
KBLDA	**71.14**	**80.05**	86.05	87.33	88.00	87.52
SKBLDA	62.52	**81.57**	**86.81**	86.57	85.86	86.86
LDA	51.00	66.09	82.14	87.95	**89.81**	**90.47**
FLDA	56.81	67.62	72.33	79.81	88.33	**90.47**
PCA-LDA	27.95	49.29	67.71	79.19	81.29	82.38
KLDA	64.33	73.81	84.48	85.71	86.67	87.52
SKLDA	44.00	71.10	**87.67**	87.10	85.14	86.86
KPCA (64.00)	26.62	36.71	49.29	51.43	51.62	52.05
SKPCA (64.14)	26.81	37.33	49.14	51.52	51.43	51.52

(a)

Dimensionality (d)	1	2	3	4	5
BLDA	69.65	80.65	**82.80**	82.55	83.15
KBLDA	**75.50**	**82.60**	**82.85**	**83.50**	**84.60**
SKBLDA	74.90	81.80	**82.65**	82.60	**84.10**
LDA	62.35	74.10	**82.50**	82.70	83.15
FLDA	60.10	71.45	75.30	76.00	83.15
PCA-LDA	39.15	60.75	78.45	78.50	78.35
KLDA	57.45	78.50	78.60	77.65	78.45
SKLDA	58.10	81.90	**82.75**	82.85	**84.10**
KPCA (77.50)	39.25	70.10	77.50	77.55	77.50
SKPCA (77.50)	39.15	70.10	77.35	77.55	77.45

(b)

Table 2. (a) Successful classification rates on the image segmentation data-set. 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.

[9] 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. 1

[10] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola and K. Müller, "Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces," IEEE Transactions on Patterns Analysis and Machine Intelligence,25(5):623-627, 2003. 2, 4

[11] 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. 6

[12] B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K. Müller, G. Rätsch, and A. Smola, "Input Space vs. Feature Space In Kernel-Based Methods," IEEE Transactions on Neural Networks, 10(5):1000-1017, 1999. 2

[13] M.E. Tipping, "Sparse Kernel Principal Component Analysis," NIPS, 633-639, 2000. 2, 4

[14] M.E. Tipping, and A.C. Faul, "Fast marginal likelihood maximisation for sparse Bayesian models," In C. M. Bishop and B. J. Frey (Eds.), Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, 2003. 4, 5

[15] V.N. Vapnik, "The Nature of Statistical Learning Theory," Springer, 1995. 2

[16] M. Zhu and A.M. Martinez, "Subclass Discriminant Analysis," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 28, No. 8, pp. 1274-1286, 2006. 1