In Proc. IEEE Workshop on Learning in Computer Vision and Patter Recognition, June 2004.

Optimal Subclass Discovery for Discriminant Analysis

Manli Zhu and Aleix M. Martínez
Depart. of Electrical and Computer Engineering
The Ohio State University
{zhum, aleix}@ece.osu.edu

Abstract

Discriminant Analysis (DA) has had a big influence in many scientific disciplines. Unfortunately, DA algorithms need to make assumptions on the type of data available and, therefore, are not applicable everywhere. For example, when the data of each class can be represented by a single Gaussian and these share a common covariance matrix, Linear Discriminant Analysis (LDA) is a good option. In other cases, other DA approaches may be preferred. And, unfortunately, there still exist applications where no DA algorithm will correctly represent reality and, therefore, unsupervised techniques, such as Principal Components Analysis (PCA), may perform better. This paper first presents a theoretical study to define when and (most importantly) why DA techniques fail (Section 2). This is then used to create a new DA algorithm that can adapt to the training data available (Sections 2 and 3). The first main component of our solution is to design a method to automatically discover the optimal set of subclasses in each class. We will show that when this is achieved, optimal results can be obtained. The second main component of our algorithm is given by our theoretical study which defines a way to rapidly select the optimal number of subclasses. We present experimental results on two applications (object categorization and face recognition) and show that our method is always comparable or superior to LDA, Direct LDA (DLDA), Nonparametric DA (NDA) and PCA.

1 Introduction: The importance of subclass divisions

Discriminant Analysis (DA) plays a central role in many research areas in science and engineering. For example, DA has been used to reduce the dimensionality of high-dimensional feature spaces to viewable 2-dimensional spaces. In economics, psychology, neuroscience and bioinformatics, DA has helped scientists to unravel useful information from otherwise meaningless feature spaces. And, of course, in computer vision, DA has been successfully used in many applications [16, 5, 2, 15, 6].

One of the most used DA algorithms is Linear Discriminant Analysis (LDA) [7, 15]. LDA is a simple algorithm that can be used for both classification and dimensionality reduction. In either case, LDA attempts to minimize the Bayes error by selecting those feature vectors $\mathbf{v}$ which maximize $\frac{|\mathbf{v}^T S_B \mathbf{v}|}{|\mathbf{v}^T S_W \mathbf{v}|}$, where S_B measures the variance between the class means, and S_W represents the variance of the samples in the same class.

Unfortunately, LDA (as well as other DA techniques) is obviously not guaranteed to work where the assumptions of the method do not hold. In such cases, unsupervised techniques, such as Principal Component Analysis (PCA), can outperform LDA and other DA approaches [14, 3, 18, 9]. For example, in some cases the data of our problem can be modelled using a single Gaussian, whereas the underlying distributions of each class cannot. In these cases, we usually prefer PCA over LDA. Should the classes correspond to linearly separable Gaussian distributions, LDA would generally be preferred. In Fig. 1(a), we show a classical example where LDA yields a more desirable result than PCA. Fig. 1(b) shows an example where PCA outperforms LDA. This discrepancy is further supported by results obtained in real applications. For example, we will show that PCA is preferred over several DA algorithms in an object categorization problem, while DA techniques are generally preferred in face recognition applications [16, 5, 2].

In this paper, we show that the drawback of DA techniques can be solved by dividing each of the classes into an adequate number of subclasses. This is illustrated in the examples shown in Fig. 1(a-b). In (a) optimal results are obtained when each class is represented using a single Gaussian. In (b) the optimal result is obtained when one of the classes is subdivided into two Gaussians.

Once each class has been subdivided by its most convenient number of subclasses, we can efficiently reduce the original space by obtaining the basis vectors of the

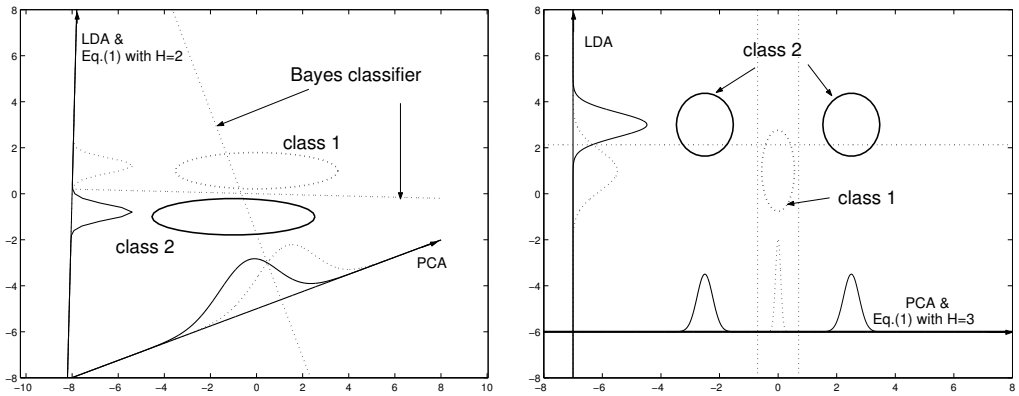


Figure 1. In (a) LDA outperforms PCA. In this example, Eq. (1) gives identical results to LDA when $H = 2$. In (b) PCA produces a most desirable result. Eq. (1) obtains the same results when one of the classes is subdivided into two subclasses, i.e. $H = 3$.

covariance matrix of the subclass means. Following the traditional notion in discriminant analysis, we will refer to this covariance matrix as the between-subclass scatter matrix, $\Sigma_B = \sum_{h=1}^{H} p_h(\mu_h - \mu)(\mu_h - \mu)^T$, where H is the total number of subclasses, p_h is the prior of subclass h, μ is the global mean, and μ_h is the mean of subclass h. Note that when each class is represented by a single subclass, $\Sigma_B = S_B$. The discriminant vectors can now be obtained by solving the following generalized eigenvalue decomposition problem [12]:

$$\Sigma_B \mathbf{V} = \Sigma_X \mathbf{V} \Lambda \,, \tag{1}$$

where Σ_X is the covariance matrix of the data, $\mathbf{V} = \{\mathbf{v}_1, \ldots, \mathbf{v}_H\}$ is the eigenvector matrix, and $\Lambda = diag\,(\lambda_1, \ldots, \lambda_H)$ is the eigenvalue matrix.

In Section 2, we study the behavior of Eq. (1) for different values of H. This will lead us to define where and *why* DA techniques fail. Based on these results, we will build a new method, Subclass Discriminant Analysis (SDA), in Sections 2 and 3. Section 4 presents our experimental results. We conclude in Section 5.

2 Subclass Divisions

In Fig. 1, we showed that Eq. (1) can (obviously) give identical results to those of LDA (when $H = C$). We will now show that, under some circumstances, the results of Eq. (1) parallel those of PCA. This latest result will show that when the i^{th} eigenvector of Σ_B is identical to the j^{th} eigenvector of Σ_X, the results of Eq. (1) (as well as the results of LDA and other DA algorithms) can become unstable.

2.1 Different values of H

The question we need to answer is the following. Is there a division of the classes into H subclasses for which the eigenvectors of Σ_B are similar to those of PCA? In Proposition 1 we will show that this can be the case, and we will define the relationship between these two results. We will show that, in most cases, several of the eigenvectors of LDA parallel those of PCA and that, in such cases, the results of Eq. (1) can become *unstable* (Theorem 1). The importance of our finding is that this is a general problem of DA and that we propose a method to solve it.

To show the relationship between PCA and Eq. (1), we will first need to prove a couple of results (Lemmas 1 and 2).

Lemma 1: If Σ_X and Σ_B have identical eigenvectors but $\Sigma_X \neq \Sigma_B$ (i.e., different eigenvalues), then the eigenvectors of PCA, $\mathbf{U}$, are related to those of Eq. (1), $\mathbf{V}$, by $\mathbf{V} = \mathbf{U}\,\mathbf{M}$, where $\mathbf{M}$ is the eigenvector matrix of

$$\left(\Lambda_X^{-1}\Lambda_B\right)\mathbf{M} = \mathbf{M}\Lambda \,,$$

Λ_B is the eigenvalue matrix of $\Sigma_B\mathbf{U} = \mathbf{U}\Lambda_B$, and Λ_X is the eigenvalue matrix of the PCA equation, $\Sigma_X\mathbf{U} = \mathbf{U}\Lambda_X$.

Proof: When Σ_B and Σ_X have proportional eigenvectors, we can assume that both share a common eigenvector matrix, $\mathbf{U} = \{\mathbf{u_1}, \mathbf{u_2}, \ldots, \mathbf{u_p}\}$, but have different eigenvalue matrices, Λ_B and Λ_X. Multiplying each of the equations above by $\mathbf{U}^T$, we obtain:

$$\Sigma_B = \mathbf{U}\Lambda_B\mathbf{U}^T, \quad and \quad \Sigma_X = \mathbf{U}\Lambda_X\mathbf{U}^T.$$

Substituting the above result in Eq. (1) we obtain

$$\mathbf{U}\Lambda_B\mathbf{U}^T\mathbf{V} = \mathbf{U}\Lambda_X\mathbf{U}^T\mathbf{V}\Lambda$$

$$\Lambda_B(\mathbf{U}^T\mathbf{V}) = \Lambda_X(\mathbf{U}^T\mathbf{V})\Lambda$$

$$\left(\Lambda_X^{-1}\Lambda_B\right)\mathbf{M} = \mathbf{M}\Lambda, \tag{2}$$

where $\mathbf{M} = \mathbf{U}^T\mathbf{V}$, which is the same as $\mathbf{V} = \mathbf{U}\,\mathbf{M}$. □

Lemma 2: Let $\Lambda_B = diag(\lambda_{B_1}, \lambda_{B_2}, \ldots, \lambda_{B_p})$, and $\Lambda_X = diag(\lambda_{X_1}, \lambda_{X_2}, \ldots, \lambda_{X_p})$ in Eq. (2), then:

- if $\frac{\lambda_{B_1}}{\lambda_{X_1}} \geq \frac{\lambda_{B_2}}{\lambda_{X_2}} \geq \ldots, \geq \frac{\lambda_{B_p}}{\lambda_{X_p}} \Rightarrow \mathbf{V} = \{\mathbf{u_1}, \ldots, \mathbf{u_p}\}$.
- if $\frac{\lambda_{B_1}}{\lambda_{X_1}} \leq \frac{\lambda_{B_2}}{\lambda_{X_2}} \leq \ldots, \leq \frac{\lambda_{B_p}}{\lambda_{X_p}} \Rightarrow \mathbf{V} = \{\mathbf{u_p}, \ldots, \mathbf{u_1}\}$.

Proof. If $\frac{\lambda_{B_1}}{\lambda_{X_1}} \geq \frac{\lambda_{B_2}}{\lambda_{X_2}} \geq \ldots, \geq \frac{\lambda_{B_p}}{\lambda_{X_p}}$, Eq. (2) has the obvious solution $\mathbf{M} = \mathbf{I}$, and therefore $\mathbf{V} = \mathbf{U}\,\mathbf{M} = \{\mathbf{u_1}, \ldots, \mathbf{u_p}\}$.

When $\frac{\lambda_{B_1}}{\lambda_{X_1}} \leq \frac{\lambda_{B_2}}{\lambda_{X_2}} \leq \ldots, \leq \frac{\lambda_{B_p}}{\lambda_{X_p}}$, we obtain

$$\mathbf{M} = \begin{pmatrix} 0 & 0 & \ldots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \ldots & 0 \\ 1 & 0 & \ldots & 0 \end{pmatrix},$$

and, therefore $\mathbf{V} = \mathbf{U}\,\mathbf{M} = \{\mathbf{u_p}, \ldots, \mathbf{u_1}\}$. □

We can now use Lemmas 1 and 2 to prove the following.

Proposition 1: Assume Σ_X and Σ_B have identical eigenvectors but $\Sigma_X \neq \Sigma_B$ (i.e., different eigenvalues), then the i^{th} eigenvector of Eq. (1), $\mathbf{v_i}$, is identical to the j^{th} eigenvector of PCA, $\mathbf{u_j}$, when $\frac{\lambda_{B_j}}{\lambda_{X_j}}$ is the i^{th} largest among $\frac{\lambda_{B_1}}{\lambda_{X_1}}, \frac{\lambda_{B_2}}{\lambda_{X_2}}, \ldots, \frac{\lambda_{B_p}}{\lambda_{X_p}}$.

Proposition 1 gives the relationship between the eigenvectors of Eq. (1) and those of PCA. In fact, we can show that when the i^{th} eigenvector of Σ_B and the j^{th} eigenvector of Σ_X ($j \leq i$) are the same, the results obtained using Eq. (1) will depend on the ratio between the eigenvalues of Σ_B and Σ_X. *Unfortunately, when this happens, there is no way of knowing which option is best for classification that given by Σ_X or that of Σ_B.* Fig. 2 illustrates this problem. In Fig. 2(a), we have three classes, each represented by a single Gaussian. We note that in this example, the first eigenvector of Σ_X and Σ_B are the same, $\mathbf{e}_1$. The problem this poses is the following. When using Eq. (1) to reduce the dimensionality of our feature space to one, we want to select a vector that has large variance according to Σ_B but small variance according to Σ_X. As we can see in Fig. 2(a), Σ_B would select $\mathbf{e}_1$ as a solution, whereas Σ_X would discourage the use of $\mathbf{e}_1$. In our example, Σ_X has a larger variance along $\mathbf{e}_1$ than Σ_B does, and, therefore, Eq. (1) will select $\mathbf{e}_2$, which is an undesirable solution.

Fig. 2(b) shows a different arrangement of the three classes. In this case, Σ_X and Σ_B are also in conflict, but the result of Eq. (1) is correct. This means that, when there is a conflict between the eigenvectors of Σ_B and those of Σ_X, the results of Eq. (1) may or may not be correct.

Theorem 1: When the i^{th} eigenvector of Σ_B is equal to one of the first i eigenvectors of Σ_X, the basis (discriminant) vectors given by Eq. (1) are not guaranteed to minimize the Bayes error for the given distributions.

The relevance of Theorem 1 is in that it defines the problem posed by the classical formulation of discriminant analysis. A classical and obvious way to minimize the problem posed by Eq. (1) is to compute the eigenvectors of Σ_B and Σ_X separately. Since Σ_B has a smaller number of vectors than Σ_X, we can first compute those eigenvectors of Σ_B which are

$$\Sigma_B\mathbf{U} = \mathbf{U}\Lambda_B\,, \tag{3}$$

then compute the eigenvectors $\mathbf{V}$ of Eq. (1) by means of the following procedure [9, 17]:

$$\begin{aligned} \mathbf{Z} &= \mathbf{U}\Lambda_B^{-1/2} \\ \mathbf{Y} &= \mathbf{Z}^T\,\Sigma_X\,\mathbf{Z} \\ \mathbf{Y}\mathbf{V}_\mathbf{Y} &= \mathbf{V}_\mathbf{Y}\Lambda_\mathbf{Y} \\ \mathbf{V} &= \Lambda_\mathbf{Y}^{-1/2}{\mathbf{V}_\mathbf{Y}}^T\mathbf{Z}^T\,. \end{aligned} \tag{4}$$

Although this algorithm may minimize the problem stated in Theorem 1 it *does not solve it* entirely. This is the reason LDA and other DA approaches fail even when the data corresponds to linearly separable Gaussian distributions.

In this paper we define a new approach to correctly address this problem. Our method is based on the results of Theorem 1 which state that the larger the number of conflicts, K, the higher the probability of obtaining an incorrect projection matrix with Eq. (1). A natural way to address this problem is thus given by those divisions of the classes into H subclasses for which this number of conflicts, K, is minimized.

Let K be the number of eigenvectors of Σ_X that are along the same direction as the eigenvectors of Σ_B. We could compute this as follows. Set $K = 0$, then:

1. for i=1 to `number of eigenvectors of` Σ_B.
2. for j=1 to i.
   If $\mathbf{u}_i \approx \mathbf{w}_j$ then $K = K + 1$;

Yet, a more accurate way of calculating the value of K is given by the following equation

$$K = \sum_{i=1}^{n_B}\sum_{j=1}^{i}(cos\theta_{i,j})^2 = \sum_{i=1}^{n_B}\sum_{j=1}^{i}\left(\mathbf{u}_i^T\mathbf{w}_j\right)^2\,, \tag{5}$$

where n_B is the number of eigenvectors of Σ_B, $\theta_{i,j}$ is the angle between the eigenvectors $\mathbf{u}_i$ and $\mathbf{w}_j$, $\mathbf{u}_i$ is the i^{th} eigenvectors of Σ_B, and $\mathbf{w}_j$ is the j^{th} eigenvector of Σ_x.

Note that Eq. (5) only considers the first i eigenvectors of Σ_X, because when Eq. (1) works properly the i^{th} eigenvector Σ_B is (in general) in accordance (i.e.,

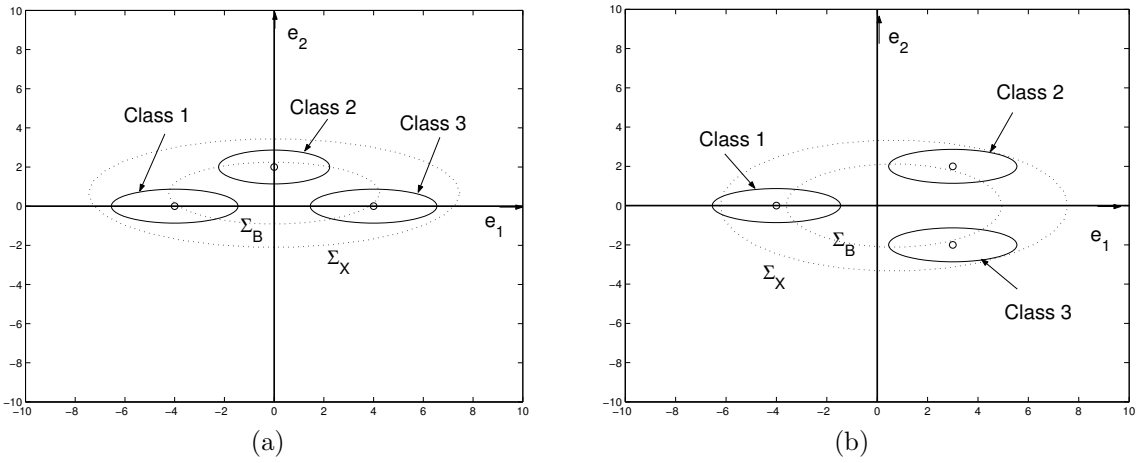


Figure 2. In (a) the results of Eq. (1) are incorrect. In (b) the results are correct.

same direction) to one of the last eigenvectors of Σ_X. Another way to estimate the correctness of our solution, would be to incorporate the ratio of the eigenvectors that agree (i.e., those having similar direction). This would give the following criteria to be maximized

$$D_H = \sum_{i=1}^{n_{B(H)}} \sum_{j=1}^{n_X} \frac{\lambda_{B(H)_i}}{\lambda_{X_j}} \left(\mathbf{u}_i^T \mathbf{w}_j\right)^2 , \qquad (6)$$

where n_X and $n_{B(H)}$ are the number of eigenvectors of Σ_X and Σ_B when the data is divided into H subclasses and $\lambda_{B(H)_i}$ is the i^{th} eigenvalue of Σ_B.

In fact, this new criteria is equivalent to our original criteria $tr\left(\Sigma_X^{-1}\Sigma_B\right)$. Therefore, maximizing Eq. (6) is equivalent to find that subspace given by Eq. (1) which maximizes the trace of the two scatter matrices.

Theorem 2: The $tr\left(\Sigma_X^{-1}\Sigma_B\right)$ is equal to

$$\sum_{i=1}^{n_{B(H)}} \sum_{j=1}^{n_X} \frac{\lambda_{B(H)_i}}{\lambda_{X_j}} \left(\mathbf{u}_i^T \mathbf{w}_j\right)^2 . \qquad (7)$$

Since, in general, the best results of Eq. (1) will be given by the data divisions, $H = j$, for which $D_j \geq D_i$, $\forall i$, we will select that division which maximizes D_H. As an alternative, we could use Eq. (5). We return to this in Section 3.2. The proof of Theorem 2 can be found in [19].

2.2 Selecting the priors: improving the recognition of similar classes

In general the priors used to compute Σ_B are $p_h = n_h/n$, where p_h is the prior of subclass h, n_h is the number of samples in subclass h, and n is the total number of samples.

However, when the total number of subclasses (H) is large, there will generally be a group of subclasses closer to each other and, hence, more susceptible to classification error. It is thus convenient to increase the value of the priors of those subclasses that are close to others and decrease the priors of those that are far from every other subclass.

An elegant way to achieve this is by using a weighting factor that measures the difficulty of classifying each pair of subclasses, (i, j). We can do that using the following between-subclass scatter matrix

$$\widetilde{\Sigma}_B \stackrel{def}{=} \sum_{i=1}^{H} \sum_{\substack{j=1 \\ j\neq i}}^{H} p_{ij}(\mu_i - \mu_j)(\mu_i - \mu_j)^T , \qquad (8)$$

where p_{ij} is the weight assigned to the pair of subclasses (i, j), and μ_i is the mean of subclass i. An important advantage of Eq. (8) is that it allows us to boost the discriminant power of those subclasses that are closer to each other by increasing the value of the corresponding weights, p_{ij}.

Since Σ_X needs to be equal to $\widetilde{\Sigma}_B + S_W$, we have to redefine our covariance as

$$\widetilde{\Sigma}_X \stackrel{def}{=} \widetilde{\Sigma}_B + S_W , \qquad (9)$$

where $S_W = \sum_{i=1}^{C} \sum_{j=1}^{n_i} (\mathbf{x}_{ij} - \mu_i)(\mathbf{x}_{ij} - \mu_i)^T$, and $\mathbf{x}_{ij}$ is the j^{th} sample of class i. The definition above, Eq. (9), is necessary to make the results of discriminant analysis stable (see Theorem 3 below).

The eigenvectors and eigenvalues of our reduced space can now be obtained by solving the generalized eigenvalue decomposition problem

$$\widetilde{\Sigma}_B \mathbf{V} = \widetilde{\Sigma}_X \mathbf{V} \Lambda . \qquad (10)$$

We can now give higher values to the weights of those pair of classes (i, j) that are closer to each other. A common way to do this, is by means of a monotonically decreasing function of the form

$$p_{ij} = \frac{n_i}{n}((\mu_i - \mu_j)(\mu_i - \mu_j)^T)^{-a},$$

where a is generally selected so that p_{ij} drops faster than $(\mu_i - \mu_j)(\mu_i - \mu_j)^T$ [13]. In our case, this number is $a = 2$.

It is important to see that the results given by Eq. (10) are still proportional to those of LDA when $H = C$ and the weights are $p_{ij} = \frac{n_i}{n}$ (i.e., we have the classical priors used by LDA). We also assume $E[\mu_i \, \mu_j^T] = E[\mu_i] \, E[\mu_j]$; i.e., knowing the mean of class i does not give information of the mean of class j.

Theorem 3: When $H = C$ and $p_{ij} = \frac{n_i}{n}$, the eigenvectors of Eq. (10) are the same as those of LDA, except for a difference in scaling given by

$$\lambda_i = \frac{2(C-1)\gamma_i}{1 + 2(C-1)\gamma_i} \, ,$$

where λ_i are the eigenvalues of Eq. (10), and γ_i are the eigenvalues of LDA.

Proof. We write our equations as

$$\begin{aligned} S_B w_i &= \gamma_i S_W w_i, \; \tilde{\Sigma}_B u_i = \delta_i S_W u_i, \\ \tilde{\Sigma}_B v_i &= \lambda_i \tilde{\Sigma}_X v_i \, . \end{aligned}$$

Since $\tilde{\Sigma}_X = \tilde{\Sigma}_B + S_W$ and $\tilde{\Sigma}_B U = S_W U \Delta$, we can now write

$$\tilde{\Sigma}_B u_i = \frac{\delta_i}{1 + \delta_i} \tilde{\Sigma}_X u_i, \; \lambda_i = \frac{\delta_i}{\delta_i + 1} \, .$$

When $p_{ij} = \frac{n_i}{n}, H = C$ and $n_i = n_j$, we have

$$\begin{aligned} \tilde{\Sigma}_B &= \sum_{i=1}^{H} \sum_{j=1}^{H} \frac{n_i}{n} (\mu_i - \mu_j)^T (\mu_i - \mu_j) \qquad (11) \\ &= \sum_{i=1}^{H} \sum_{j=1}^{H} \frac{n_i}{n} [(\mu_i - \mu) - (\mu_j - \mu)]^T [(\mu_i - \mu) - (\mu_j - \mu)] \\ &= 2 \sum_{j=1}^{H} S_B - 2 \sum_{i=1}^{H} \frac{n_i}{n} (\mu_i - \mu)^T (\mu_i - \mu) \\ &= 2(C-1) S_B \, . \end{aligned}$$

And, rearranging terms $\tilde{\Sigma}_B U = S_W U \Delta$ can be written as $2(C-1) S_B U = S_W U \Delta$, which is the same as

$$S_B u_i = \frac{\delta_i}{2(C-1)} S_W u_i, \; \gamma_i = \frac{\delta_i}{2(C-1)} \, .$$

Combining these results, we get

$$\lambda_i = \frac{2(C-1)\gamma_i}{1 + 2(C-1)\gamma_i} \, . \qquad (12)$$

□

This result guarantees that Eq. (10) will at least be as good as LDA. Should the results be superior for other values of H, we will obviously obtain better discriminant spaces.

3 Subclass Discriminant Analysis

In the section above, we showed that one can achieve optimal feature extraction and classification results by simply dividing each of the classes into an adequate number of subclasses, H. To accomplish this though, one needs to solve two problem. The first one is to find ("discover") the subclasses of each class that best divide the data. The second problem is that of selecting the partition which gives optimal classification results.

For computational reasons, it is quite obvious that we cannot test every possible division of n samples into H subclasses for every possible value of H. However, when these subclasses are assumed to correspond to Gaussian distributions, *it is reasonable to expect the samples of each subclass to be close to each other*. The method described below takes advantage of this fact to reduce the computational burden of our approach. We will also use the results of Theorem 2 to select that value of H for which we expect better discrimination.

3.1 NN clustering

This simple procedure (Nearest Neighbor-based) sorts the vectors of each class, $\mathbf{x}_{i,1}, \ldots, \mathbf{x}_{i,n_c}$ (where n_c is the number of samples in class c), as follows. $\mathbf{x}_{i,1}$ and $\mathbf{x}_{i,n_c}$ are the two most distant feature vectors of class i (i.e., the Euclidean distance between these two vectors is the largest: $argmax_{j,k} \|\mathbf{x}_{i,j} - \mathbf{x}_{i,k}\|$). $\mathbf{x}_{i,2}$ is the closest feature vector to $\mathbf{x}_{i,1}$; and $\mathbf{x}_{i,n_c-1}$ is the closest to $\mathbf{x}_{i,n_c}$. In general, $\mathbf{x}_{i,j}$ is the $j - 1^{th}$ feature vector closest to $\mathbf{x}_{i,1}$, and $\mathbf{x}_{i,n_c-j}$ is the j^{th} closest to $\mathbf{x}_{i,n_c}$.

The procedure described above is useful when H is larger than C. For example, if $H = 2C$, then each class is divided into two groups (i.e., two subclasses). This is suitable for those cases where *i)* the underlying distribution of each of the classes is not Gaussian, but can be represented as a combination of two or more Gaussians, or *ii)* the classes are not separable, but the subclasses are.

Obviously, there are many ways in which the data of each class can be divided into subclasses. For computational efficiency, we will assume that each subclass has the same number of samples. We have, nonetheless, experimented with the *K-means* clustering algorithm and the nonparametric valley-seeking algorithm of Koontz and Fukunaga [9] where this restriction does not apply. The classification results obtained with these two methods were, however, comparable; yet the computational cost was higher. Based on this observation, we have decided to work with the NN approach described in this section. Note that other algorithms, such as mixture models, could not generally be efficiently used because, in many cases, we would not have enough training samples per class to successfully run such algorithms.

3.2 Optimal value of H

Finally, we need to define a way to choose the optimal value of H. To do this, we will use the result given in Theorem 2 and select that value of H that minimizes D_H. Note, however, that the larger the value of H, the larger the number of eigenvectors in Σ_B. To prevent our algorithm to be bias toward those representations with larger values of H, we will normalized our results by the number of eigenvalues used. A solution to this problem is to normalize the eigenvalues such that they sum up to one. Our optimal number of classes, H_o, is then given by

$$H_o = argmax_h \, \widetilde{D}_H \, , \tag{13}$$

where $\widetilde{D}_H$ is

$$\widetilde{D}_H = \sum_{i=1}^{n_{B(H)}} \sum_{j=1}^{n_X} \frac{\widetilde{\lambda}_{B(H)_i}}{\lambda_{X_j}} \left(\mathbf{u}_i^T \mathbf{w}_j\right)^2 \, , \tag{14}$$

and $\widetilde{\lambda}_{B(H)_i}$ are the normalized eigenvalues of Σ_B such that

$$\sum_{i=1}^{n_{B(H)}} \widetilde{\lambda}_{B(h)_i} = 1.$$

4 Experimental Results

DA approaches have been largely applied to the classification of images of objects. We will now show how our algorithm applies to two particular problems: *a)* object categorization, and *b)* face recognition. We will compare our results to LDA, Direct LDA (DLDA) [17], Nonparametric DA (NDA) [9], and PCA.

4.1 Object categorization

A classical problem in computer vision is to classify a set of objects into a group of known categories (e.g., apples, cars, etc.).

In our experiments, we have used the ETH-80 database which contains several images of the following categories: apples, pears, cars, cows, horses, dogs, tomatoes, and cups [11]. Each category includes the images of ten objects (e.g., ten different pears) photographed at a total of 41 orientations. This gives us a total of 410 images per category. Four images of four different cows are shown in Fig. 3(a-d).

Feature-based classification

In this approach, we first obtain the silhouette of the object (i.e., the contour that separates the object from the background), Fig. 3(f). We then compute the centroid of this silhouette and, finally, calculate the length of equally separated lines that connect the centroid with the silhouette; see Fig. 3(g). In our experiments we obtain a total of 300 distances, which gives us a feature space of 300 dimensions.

We now use our method (SDA) as well as LDA, DLDA, NDA, and PCA to first reduce the dimensionality of our feature space. Then, we use the nearest neighbor algorithm for classification in these reduced spaces. To test the classification accuracy of each method, we used the leave-one-object out test. To do this, we use 79 of the 80 objects for training and one for testing. Since there are obviously 80 ways in which we can leave one object out, we repeated the test eighty times and computed the mean and standard deviation. The results are summarized in Fig. 4(a).

Since we used the leave-one-object out for testing, the value of H_o varied each time. The average value of H_o was 72, which means we would have an average of about 45 samples per subclass.

4.2 Face recognition

In this experiment, we used the AR-face database [14]. The AR-face database consists of frontal-view face images of over 100 people. Each person was photographed under different lighting conditions and distinct facial expressions, and some images have partial occlusions (sunglasses or scarf). A total of 13 images were taken in each sessions for a total of two sessions; i.e., 26 images per person. These sessions were separated by an interval of two weeks.

We used the first 13 images (corresponding to the first session) of 100 randomly selected individuals for training; i.e., $1,300$ training images. The 13 images of the second session (for the same people) were used for testing. This problem is known as the *recognition of duplicates* in the face recognition literature.

Appearance-based classification

We now use an appearance-based approach for recognition. We first crop the face part of the image and then resize all images to a standard image size of 17 by 12 pixels, which yields a 204-dimensional feature space. During the cropping process we align all faces to have a common position for the eyes and mouth (since this has been shown to improve accuracy).

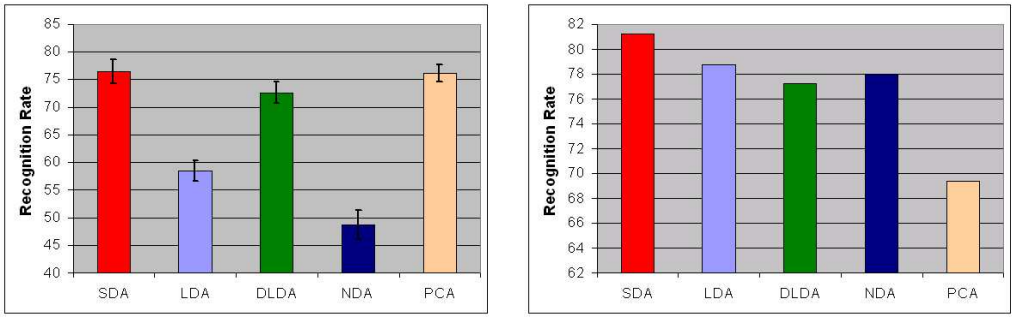
The results obtained using our algorithm (SDA) as well as those of LDA, DLDA, NDA and PCA are shown in Fig. 4(c). In this case $H_o = 300$.

5 Discussion and Conclusions

We have theoretically shown that to optimally separate classes, one needs to first subdivide the sample of each class into an adequate number of subclasses. Our result is supported by intuitive ideas (see Fig. 1)

(a) (b) (c) (d) (e) (f) (g)

Figure 3. (a-d) Four examples of four different cows as seen from the same viewing angle. (f) The silhouette of the cow shown in (e). (g) Each line measures the distance from the centroid to several points in the silhouette.



(a) (b)

Figure 4. (a) Results of the categorization problem using the feature-based approach. (b) Face recognition results using an appearance-based approach.

and by our experimental results shown above. An additional advantage of SDA is that it allows to compute a larger number of eigenvectors than most DA algorithms, because the rank of Σ_B is usually larger than the rank of S_B. We have also shown that PCA will outperform most DA techniques in some applications, but not SDA. However, other unsupervised techniques such as ICA (Independent Components Analysis) [10, 1] may be superior to our SDA algorithm as defined above. We can nonetheless address this problem by incorporating higher-moments in our formulation

5.1 Higher-moments

Fig. 5 shows a classical example where ICA [1] and other techniques such as SAVE [4] outperform PCA. In this example, LDA could not be computed because the global mean and the class means are the same; i.e. $S_B = 0$. It should also be clear by now that if each of the classes is not subdivided into two or more subclasses, our previous equation, Eq. (10), would have the same shortcomings as LDA. When each class is, however, subdivided into subclasses, SDA will output a result similar to PCA. In this case, the results of SDA is unfortunately incorrect.

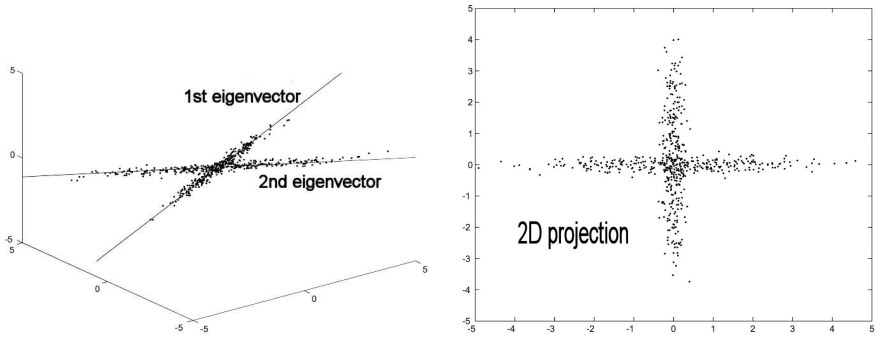
To correctly address this problem, we need to extend our previous equation to include higher moments of the data. We can achieve this by exchanging the between-subclass scatter matrix, Σ_B, with a more adequate matrix such as

$$\Sigma_\rho = \sum_{h=1}^{H} p_h \left(E[\Sigma_h] - \Sigma_h\right)^2 , \qquad (15)$$

where Σ_h is the covariance matrix of subclass h and $E[\mathbf{X}]$ is the expected vector of $\mathbf{X}$ (i.e. mean). This would extend the use of SDA to those applications where PCA and DA techniques do not perform well. Our current efforts are along these directions.

5.2 Conclusions

DA approaches have proven to be useful in many areas of research. Unfortunately, due to the assumptions embedded in their formulation, DA approaches fail in many applications. In this contribution, we have first studied the limitations and drawbacks of the classical equation of DA and then proposed a new algorithm to overcome these problems. Our experimental results confirm our claims. In three recognition experiments, the results of our algorithm were either superior or comparable to other DA approaches (e.g., LDA, DLDA and NDA) and to unsupervised techniques (e.g., PCA). Extensions of this approach to include higher-moments of the data are currently underway.



(a) (b)

Figure 5. In this example, we have two classes, each corresponding to a Gaussian distribution. The two Gaussians cross each other at their mean, and are only separated by a small angle. ICA and SAVE are able to detect the cross structure of the two classes. SDA would also be able to obtained this information if we used Eq (15) in place of Σ_B.

Acknowledgments

We would like to thank the reviewers for their constructive comments. This research was partially supported by NIH.

References

[1] M.S. Bartlett, "Face Image Analysis by Unsupervised Learning," Kluwer International Series on Engineering and Computer Science, Vol. 612, Kluwer Academic, 2001.

[2] P.N. Belhumeur, J.P. Hespanha and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection," IEEE Trans. on Pattern Analysis and Machine Intelligence 19(7):711-720, 1997.

[3] R. Beveridge, K. She, B. Draper, G. Givens, "A Nonparametric Statistical Comparison of Principal Component and Linear Discriminant Subspaces for Face Recognition," In Proc. Computer Vision and Pattern Recognition, Kauai, HI, pp. I:535-542, 2001.

[4] R.D. Cook and S. Weisberg, "Sliced Inverse Regression for Dimensionality Reduction: Comment," J. Am. Statistical Soc. 86(414):328-332, 1991.

[5] K. Etemad and R. Chellapa, "Discriminant Analysis for Recognition of Human Face Images," Journal of Optical Society of American A 14(8):1724-1733, 1997.

[6] W.J. Ewans and G.R. Grant, "Statistical Methods in Bioinformatics," Springer-Verlag, 2001.

[7] R.A. Fisher, "The Statistical Utilization of Multiple Measurements," Annals of Eugenics, 8:376-386, 1938.

[8] J.H. Friedman, "Regularized Discriminant Analysis," J. Am. Statistical Assoc. 84, 165-175, 1989.

[9] K. Fukunaga, "Introduction to Statistical Pattern Recognition (2nd edition)," Academic Press, 1990.

[10] C. Jutten and J. Herault,, "Blind Separation of Sources I: An adaptive algorithm based on neuromimetic architecture," Signal Processing 24(1):1-10, 1991.

[11] B. Leibe and B. Schiele, "Analyzing Appearance and Contour Based Methods for Object Categorization," In Proc. IEEE Computer Vision and Pattern Recognition, Madison (WI), June 2003.

[12] J. Li, "Sliced Inverse Regression for Dimensionality Reduction," J. Am. Stat. Soc. 86(414):316-327, 1991.

[13] R. Lotlikar and R. Kothari, "Fractional-Step Dimensionality reduction," IEEE Trans. on Pattern Analysis and Machine Intelligence 22(6):623-627, 2000.

[14] A.M. Martínez and A.C. Kak, "PCA versus LDA," IEEE Trans. Pattern Analysis and Machine Intelligence 23(2):228-233, 2001.

[15] C.R. Rao, "Prediction of Future Observation in Growth Curve Models," Stat. Science 2:434-471, 1987.

[16] D.L. Swets and J.J. Weng, "Using Discriminant Eigenfeatures for Image Retrieval," IEEE Trans. on Pattern Analysis and Machine Intelligence 18(8):831-836, 1996.

[17] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data – with applications to face recognition," Pattern Recognition 34:2067-2070, 2001.

[18] M. Zhu, A.M. Martinez and H. Tan, "Template-based Recognition of Sitting Postures," In Proc. IEEE Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction, Madison (WI), 2003.

[19] M. Zhu and A.M. Martinez, "An Introduction to Subclass Discriminant Analysis," OSU Tech. Rep., 2004.