

# Where Are Linear Feature Extraction Methods Applicable?

Aleix M. Martínez, *Member, IEEE*, and Manli Zhu

**Abstract**—A fundamental problem in computer vision and pattern recognition is to determine where and, most importantly, why a given technique is applicable. This is not only necessary because it helps us decide which techniques to apply at each given time. Knowing why current algorithms cannot be applied facilitates the design of new algorithms robust to such problems. In this paper, we report on a theoretical study that demonstrates where and why generalized eigen-based linear equations do not work. In particular, we show that when the smallest angle between the  $i$ th eigenvector given by the metric to be maximized and the first  $i$  eigenvectors given by the metric to be minimized is close to zero, our results are not guaranteed to be correct. Several properties of such models are also presented. For illustration, we concentrate on the classical applications of classification and feature extraction. We also show how we can use our findings to design more robust algorithms. We conclude with a discussion on the broader impacts of our results.

**Index Terms**—Feature extraction, generalized eigenvalue decomposition, performance evaluation, classifiers, pattern recognition.

## 1 INTRODUCTION

IN recent years, linear methods have played a major role in the study of large quantities of data. Primarily, to find which features best describe a class (e.g., features that best represent cars), and to build classifiers able to assign class labels to new feature vectors (e.g., cars versus houses) [28], [13], [10], [16]. Linear methods have predominated because, when the underlying distributions of the classes are linearly separable, a small number of samples usually suffices to perform the above mentioned tasks, and because most problems can be formulated as an eigenvalue decomposition equation for which many efficient algorithms exist [11]. In this paper, we show that even when the classes are linearly separable, the results of such linear methods are not guaranteed to be correct. We present a simple mechanism to test the validity of the results of such linear methods and define a simple robust algorithm to compute the results of eigen-based equations where our assumptions hold. In this study, a geometric interpretation of eigen-based linear methods is presented and used as a tool for the design of the robust algorithm. Several generalization properties of the linear methods studied are also outlined.

The aforementioned tasks, namely, feature extraction (i.e., where a set of features or linear combination of them are chosen to represent each class) and classification (i.e., where the feature space is linearly divided into a set of regions belonging to distinct classes), can be readily computed from the following generalized eigenvalue decomposition problem

$$\mathbf{M}_W \mathbf{V} = \mathbf{M}_U \mathbf{V} \Lambda, \quad (1)$$

where  $\mathbf{M}_W$  and  $\mathbf{M}_U$  are the metric matrices that define the measure to be maximized and that to be minimized,

- The authors are with the Department of Electrical and Computer Engineering, 205 Dreese Lab, 2015 Neil Ave., The Ohio State University, Columbus, OH 43210. E-mail: {aleix, zhumi}@ece.osu.edu.

Manuscript received 20 Oct. 2004; revised 4 May 2005; accepted 5 May 2005; published online 13 Oct. 2005.

Recommended for acceptance by J. Buhmann.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0569-1004.

respectively, [28], [17], [10]. Finding the eigenvectors,  $\mathbf{v}$ , of (1) is therefore equivalent to selecting those vectors which maximize

$$\frac{|\mathbf{v}^T \mathbf{M}_W \mathbf{v}|}{|\mathbf{v}^T \mathbf{M}_U \mathbf{v}|}. \quad (2)$$

Here, we assume that  $\mathbf{M}_W$  and  $\mathbf{M}_U$  are always symmetric and positive-defined which is necessary for these matrices to define a metric.

Equation (1) has been used to define many feature extraction algorithms, both for classification and regression. A well-known example is linear discriminant analysis (LDA) [28], [24]. In this case,  $\mathbf{M}_W$  is defined as the sample between-class scatter matrix (i.e., the covariance of the class means),

$$\mathbf{M}_W = \mathbf{S}_B = \sum_{i=1}^c (\mu_i - \mu)(\mu_i - \mu)^T, \quad (3)$$

where  $c$  is the number of classes,  $\mu_i$  the sample mean of class  $i$  and  $\mu$  the global mean (including the samples of all classes). A first option for  $\mathbf{M}_U$  is the within-class scatter matrix (i.e., the sample mean covariance matrix of every class), defined as

$$\mathbf{S}_W = \frac{1}{n} \sum_{i=1}^c \Sigma_i, \quad (4)$$

where  $\Sigma_i$  is the covariance matrix of the samples in class  $i$  and  $n$  and  $c$  are the number of samples and classes, respectively. Because minimizing the sample mean covariance matrix of all classes is equivalent to the minimization of the covariance of the data, a well-known alternative for  $\mathbf{M}_U$  is the sample covariance matrix,  $\Sigma_X$  [10]. The sample covariance matrix is defined as

$$\Sigma_X = \frac{1}{n} \sum_{i=1}^n (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)^T, \quad (5)$$

where  $\mathbf{x}_i$  represent the  $i$ th sample vectors.

Figs. 1a and 1d show classical examples where LDA is known to successfully extract the most discriminant feature,

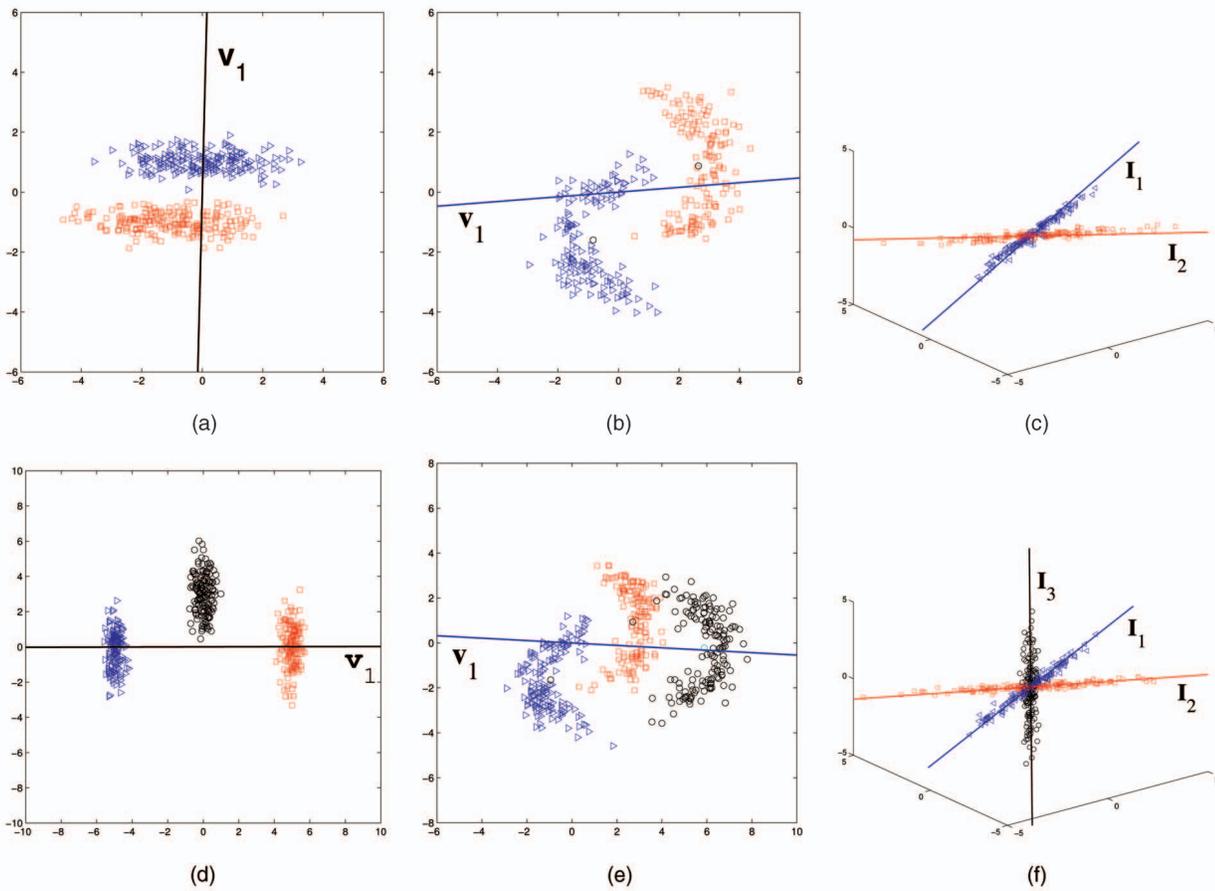


Fig. 1. This figure shows three two-class and three three-class examples. (a) Shows the first and optimal (in the sense of minimizing the Bayes error) eigenvector,  $\mathbf{v}_1$ , of LDA. (b) Shows the first and optimal eigenvector of NDA. (c) The two independent components that best describe the classes (in the sense of maximizing independence between the samples of different classes),  $\mathbf{I}_1$  and  $\mathbf{I}_2$ , are orthogonal to those of (1) when  $\mathbf{M}_W = \Sigma_\rho$  and  $\mathbf{M}_U = \Sigma_X$ . (d), (e), and (f) Do the same, but for the three-class cases.

$\mathbf{v}_1$ , from a simple two-dimensional space. These results were obtained by first generating a large number of samples from each of the two distribution and using  $\mathbf{M}_W = \mathbf{S}_B$  and  $\mathbf{M}_U = \mathbf{S}_W$  as metrics in (1).

Another successful application of (1), which is also well-known within the computer vision community, is Non-parametric Discriminant Analysis (NDA) [9], [10]. In this second case,  $\mathbf{M}_W$  is the sample nonparametric between-class scatter matrix, given by

$$\mathbf{S}_B = \frac{1}{n} \sum_{i=1}^n \alpha_i (\mathbf{x}_i - \mathcal{M}_i)(\mathbf{x}_i - \mathcal{M}_i)^T, \quad (6)$$

where  $\mathcal{M}_i$  is the mean of the  $k$  nearest samples to  $\mathbf{x}_i$  belonging to a different class than that of  $\mathbf{x}_i$  and  $\alpha_i$  is any scale factor which prevents the results to be affected by isolated samples located far from the rest.

As in LDA, NDA has two options for  $\mathbf{M}_U$ , the within-class scatter matrix,  $\mathbf{S}_W$ , or the sample covariance matrix,  $\Sigma_X$ . In Figs. 1b and 1e, we show two examples where NDA is known to successfully obtain the most discriminant feature vector for the given distributions. Again, the result of NDA is represented by the vector  $\mathbf{v}_1$ .

We have recently shown that (1) can also be used to obtain the independent components of the data [36]. These are orthogonal or identical to the basis vectors obtained

with (1) when  $\mathbf{M}_W$  estimates the variance between the covariance matrix of every class, i.e.,

$$\mathbf{M}_W = \Sigma_\rho = \sum_{i=1}^c (E[\Sigma_i] - \Sigma_i)^2, \quad (7)$$

where  $E[\Sigma_i]$  is the expected (mean) sample covariance matrix. Since discrimination usually requires to minimize the distance among the samples of the same class, the covariance matrix is generally assign as metric in  $\mathbf{M}_U$ .

Figs. 1c and 1f show the independent components of the data,  $\mathbf{I}_i$ , recovered using the approach defined in the preceding paragraph. Note that the extraction of these feature vectors is necessary if one wants to project the data of the two classes in a reduced space where the samples of each class collapse in a small area and those of different classes fall far apart. The discriminant vectors where this happens are either orthogonal or equal to each  $\mathbf{I}_i$ . We should also point out that neither LDA or NDA would be able to successfully obtain such results [1].

Of course, (1) is not limited to the three examples defined above. For example, in metric-based Principal Component Analysis (PCA) [31],  $\mathbf{M}_W$  is the sample covariance matrix and  $\mathbf{M}_U$  is the covariance matrix of errors or uncertainties. In spatiotemporal applications, the goal is to maximize the signal to noise ratio which can be modeled as  $\mathbf{M}_W$  being the vectors of known signals at time  $t$  and  $\mathbf{M}_U$  the spatiotemporal covariance matrix of the noise term. Linear regression

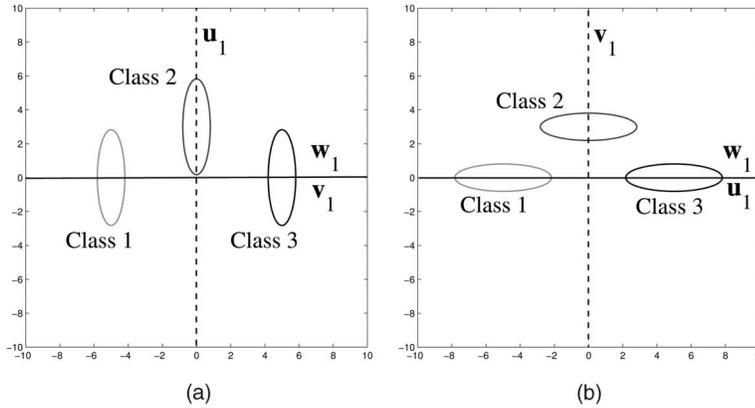


Fig. 2. (a) A classical example where LDA performs as expected. Note that the angle between the first eigenvectors of  $\mathbf{M}_W$  and  $\mathbf{M}_U$  is large. (b) An example where LDA fails. In this second case, the angle between the first eigenvector of each of the matrices defining the metrics to be maximized and minimized is zero. In (a) and (b),  $\mathbf{M}_W = \mathbf{S}_B$  and  $\mathbf{M}_U = \mathbf{S}_W$ .

approaches have also been successfully formulated within this framework [20], [5]. And among many others, (1) has also been used in redundancy analysis, canonical correlate analysis, and principal components of instrumental variables [25], [22], [17], [28].

In computer vision, linear methods that are either based or inspired on equations similar to (1) have also flourished. These may be slightly different to those defined above, but our results should be extendable to several of these methods as well.

To date, techniques based on (1) are assumed to generally work for homoscedastic data, where all classes share a common covariance matrix (i.e.,  $\Sigma_i = \Sigma_j, \forall i, j$ ) but have distinct means (i.e.,  $\mu_i \neq \mu_j, \forall i \neq j$ ). Experimental results, however, show otherwise. Equation (1) has been successfully applied to heteroscedastic data and, most importantly, has sometimes failed when applied to homoscedastic data. It is, however, unknown when and why such models do not work.

In the following, we show that the success of the linear methods defined in (1) does not depend on whether the data is homoscedastic or not. We will demonstrate that the results of (1) are not guaranteed to be correct when the smallest angle between the  $i$ th eigenvector of  $\mathbf{M}_W$  and the first  $i$  eigenvectors of  $\mathbf{M}_U$  is close to zero. This problem is illustrated in Fig. 2 where we compare the results of LDA on two types of data. The first example (shown in Fig. 2a) is that previously illustrated in Fig. 1d, for which LDA is known to perform well. In contrast, our second example (Fig. 2b) shows an unsuccessful case for LDA. Note from Fig. 2 that while in Fig. 2a the angle between the eigenvectors of  $\mathbf{M}_W$  and  $\mathbf{M}_U$  ( $\mathbf{w}_1$  and  $\mathbf{u}_1$  in the figure) is relatively large, in the example of Fig. 2b the angle between the first eigenvector of  $\mathbf{M}_W$  and the first of  $\mathbf{M}_U$  is zero and the results are incorrect. This can be nicely summarized as follows:

**Theorem 1.** Let  $\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_q\}$  and  $\Lambda_W = \{\lambda_{W_1}, \dots, \lambda_{W_q}\}$  be the eigenvectors and eigenvalues of  $\mathbf{M}_W \mathbf{W} = \mathbf{W} \Lambda_W$ , and  $\mathbf{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_p\}$  and  $\Lambda_U = \{\lambda_{U_1}, \dots, \lambda_{U_p}\}$  the eigenvectors and eigenvalues of  $\mathbf{M}_U \mathbf{U} = \mathbf{U} \Lambda_U$ , where  $q$  and  $p$  are the ranks of  $\mathbf{M}_W$  and  $\mathbf{M}_U$ , respectively,  $\lambda_{W_1} \geq \lambda_{W_2} \geq \dots \geq \lambda_{W_q}$ ,  $\lambda_{U_1} \geq \lambda_{U_2} \geq \dots \geq \lambda_{U_p}$ , and  $p \geq q$ . Define

$$K = \sum_{i=1}^r \sum_{j=1}^i (\cos \theta_{i,j})^2 = \sum_{i=1}^r \sum_{j=1}^i (\mathbf{u}_j^T \mathbf{w}_i)^2, \quad (8)$$

where  $r \leq q$ , and  $\theta_{i,j}$  is the angle between the eigenvectors  $\mathbf{w}_i$  and  $\mathbf{u}_j$ . Then, if  $K > 0$  the basis vectors given by (1) are not guaranteed to minimize the Bayes error for the given distributions of the data.

## 2 A GEOMETRIC INTERPRETATION OF (1)

Before we give a formal proof of the relationship between (1) and (8), it is easier to show the geometric implications of Theorem 1; which may also serve as an intuitive proof.

As discussed above, Fig. 2 shows the eigenvectors of  $\mathbf{M}_W$  and  $\mathbf{M}_U$  for the LDA algorithm in two different examples. In Fig. 2a, the angle between the first eigenvector of  $\mathbf{M}_W$  (i.e., the first eigenvector of  $\mathbf{S}_B$ ,  $\mathbf{w}_1$ ) and the first of  $\mathbf{M}_U$ ,  $\mathbf{u}_1$ , is relatively large. Hence, the value of  $K$  will be small, which is an indication of good generalization and, if the Bayes error is small, of good predictability.

For comparison, we also showed results on a problem where LDA does not perform as expected. This is in Fig. 2b. In this example, there are three classes, each represented by a single Gaussian with identical covariance matrix (i.e., homoscedastic data). We see that the basis vector generated by LDA,  $\mathbf{v}_1$ , does not minimize the Bayes error. In fact, the optimal result would be the one orthogonal to  $\mathbf{v}_1$ .

To see why this happens, it is necessary to go back to the definition given in (2). According to that result, the goal is to maximize the measure given by  $\mathbf{M}_W$ , while minimizing that of  $\mathbf{M}_U$ . This works well when the first eigenvector of  $\mathbf{M}_W$  and the first eigenvector of  $\mathbf{M}_U$  are orthogonal to each other. However, when the  $i$ th eigenvector of  $\mathbf{M}_W$  and the  $j$ th eigenvector of  $\mathbf{M}_U$  (for any  $j \leq i$ ) are the same, the results obtained using (1) will depend on the ratio between the eigenvalues of  $\mathbf{M}_W$  and  $\mathbf{M}_U$ . And, unfortunately, in these cases, there is no way of knowing which option is best for classification that given by  $\mathbf{M}_W$  or that of  $\mathbf{M}_U$ . Fig. 2b illustrates this. We note that in this example, the first eigenvector of  $\mathbf{M}_W$  and  $\mathbf{M}_U$  are the same (i.e.,  $\mathbf{e}_1 = \mathbf{w}_1 = \mathbf{u}_1$ ) and, therefore, when we use (1) to reduce the dimensionality of our feature space to one, we cannot select a vector that has large variance according to  $\mathbf{M}_W$  and small

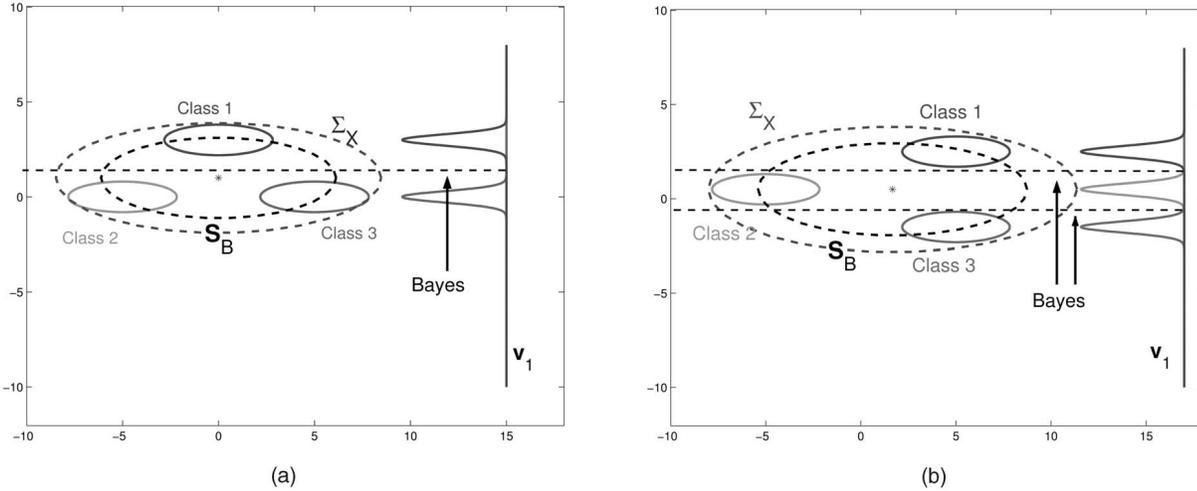


Fig. 3. This figure shows that when  $\mathbf{u}_1 = \mathbf{w}_1$  the algorithms defined by (1) become unstable and may not minimize the Bayes error for the given data distributions. In these two examples, we used the metrics of LDA defined above. (a) Equation (1) does not minimize the Bayes error of the given distributions. (b) Although in this second example  $\mathbf{u}_1$  is also equal to  $\mathbf{w}_1$ , the result given by (1) is correct.

according to  $\mathbf{M}_U$ . As we can see in the figure,  $\mathbf{M}_W$  would like to select  $\mathbf{e}_1$  as a solution, whereas  $\mathbf{M}_U$  would discourage the use of  $\mathbf{e}_1$ . In our example,  $\mathbf{M}_U$  has a larger variance along  $\mathbf{e}_1$  than  $\mathbf{M}_W$  does and, therefore, (1) will select the vector orthogonal to  $\mathbf{e}_1$  (i.e.  $\mathbf{u}_2$ ) as the solution. This is obviously an undesirable result; see Fig. 3a.

We refer to this problem as the existence of a *conflict* between the eigenvectors of  $\mathbf{M}_W$  and  $\mathbf{M}_U$ . More formally, a measure of this *conflict* is given by (8). Low values of  $K$  indicate there is no or little conflict. High values mean the results of (1) need not be correct.

To formally prove this result, we first note that the discriminant power of (1),

$$\text{tr}(\mathbf{M}_U^{-1}\mathbf{M}_W), \quad (9)$$

can also be expressed in the form of the geometric interpretation given above in (8).

**Theorem 2.** *The  $\text{tr}(\mathbf{M}_U^{-1}\mathbf{M}_W)$  is equal to*

$$\sum_{i=1}^q \sum_{j=1}^p \frac{\lambda_{W_i}}{\lambda_{U_j}} (\mathbf{u}_j^T \mathbf{w}_i)^2. \quad (10)$$

**Proof.** Since

$$\mathbf{M}_U = \mathbf{U}\mathbf{\Lambda}_U\mathbf{U}^T = \mathbf{U}\mathbf{\Lambda}_U^{1/2}\mathbf{\Lambda}_U^{1/2}\mathbf{U}^T = (\mathbf{U}\mathbf{\Lambda}_U^{1/2})(\mathbf{U}\mathbf{\Lambda}_U^{1/2})^T,$$

we can rework (1) as follows:

$$\begin{aligned} \mathbf{M}_W\mathbf{V} &= (\mathbf{U}\mathbf{\Lambda}_U^{1/2})(\mathbf{U}\mathbf{\Lambda}_U^{1/2})^T\mathbf{V}\mathbf{\Lambda} \\ (\mathbf{U}\mathbf{\Lambda}_U^{1/2})^{-1}\mathbf{M}_W\left[(\mathbf{U}\mathbf{\Lambda}_U^{1/2})^T\right]^{-1}(\mathbf{U}\mathbf{\Lambda}_U^{1/2})^T\mathbf{V} &= (\mathbf{U}\mathbf{\Lambda}_U^{1/2})^T\mathbf{V}\mathbf{\Lambda}. \end{aligned}$$

Now, let  $\mathbf{Y} = (\mathbf{U}\mathbf{\Lambda}_U^{1/2})^T\mathbf{V}$ . And, since  $\mathbf{U}$  is a matrix of orthonormal vectors, we can simplify the above equation to

$$\mathbf{\Lambda}_U^{-1/2}\mathbf{U}^T\mathbf{M}_W\mathbf{U}\mathbf{\Lambda}_U^{-1/2}\mathbf{Y} = \mathbf{Y}\mathbf{\Lambda}. \quad (11)$$

Equations (1) and (11) have the same eigenvalues  $\Lambda$ , but different eigenvectors, which are related by  $\mathbf{Y} = (\mathbf{U}\mathbf{\Lambda}_U^{1/2})^T\mathbf{V}$ .

$$\mathbf{\Lambda}_U^{-1/2} \begin{pmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_p^T \end{pmatrix} \left( \sum_{i=1}^{q-1} \lambda_{W_i} \mathbf{w}_i \mathbf{w}_i^T \right) \mathbf{U}\mathbf{\Lambda}_U^{-1/2}\mathbf{Y} = \mathbf{Y}\mathbf{\Lambda}$$

$$\mathbf{\Lambda}_U^{-1/2} \begin{pmatrix} \sum_{i=1}^{q-1} \lambda_{W_i} \langle \mathbf{u}_1, \mathbf{w}_i \rangle \mathbf{w}_i^T \\ \cdots \\ \sum_{i=1}^{q-1} \lambda_{W_i} \langle \mathbf{u}_p, \mathbf{w}_i \rangle \mathbf{w}_i^T \end{pmatrix} (\mathbf{u}_1, \dots, \mathbf{u}_p) \mathbf{\Lambda}_U^{-1/2}\mathbf{Y} = \mathbf{Y}\mathbf{\Lambda}$$

$$\begin{pmatrix} \sum_{i=1}^{q-1} \frac{\lambda_{W_i}}{\sqrt{\lambda_{U_1}\lambda_{U_1}}} \langle \mathbf{u}_1, \mathbf{w}_i \rangle \mathbf{w}_i^T \mathbf{u}_1 & \cdots & \sum_{i=1}^{q-1} \frac{\lambda_{W_i}}{\sqrt{\lambda_{U_1}\lambda_{U_p}}} \langle \mathbf{u}_1, \mathbf{w}_i \rangle \mathbf{w}_i^T \mathbf{u}_p \\ \cdots & \cdots & \cdots \\ \sum_{i=1}^{q-1} \frac{\lambda_{W_i}}{\sqrt{\lambda_{U_p}\lambda_{U_1}}} \langle \mathbf{u}_p, \mathbf{w}_i \rangle \mathbf{w}_i^T \mathbf{u}_1 & \cdots & \sum_{i=1}^{q-1} \frac{\lambda_{W_i}}{\sqrt{\lambda_{U_p}\lambda_{U_p}}} \langle \mathbf{u}_p, \mathbf{w}_i \rangle \mathbf{w}_i^T \mathbf{u}_p \end{pmatrix} \mathbf{Y} = \mathbf{Y}\mathbf{\Lambda},$$

where  $\langle \mathbf{a}, \mathbf{b} \rangle$  is the inner product between  $\mathbf{a}$  and  $\mathbf{b}$ .

For any matrix  $\mathbf{M}$  (of  $p$  rows and  $p$  columns) the  $\text{tr}(\mathbf{M}) = \sum_{i=1}^p \lambda_i$  and, therefore,

$$\begin{aligned} \text{tr}(\mathbf{M}_U^{-1}\mathbf{M}_W) &= \sum_{j=1}^p \lambda_j \\ &= \sum_{j=1}^p \sum_{i=1}^q \frac{\lambda_{W_i}}{\sqrt{\lambda_{U_j}\lambda_{U_j}}} \langle \mathbf{u}_j, \mathbf{w}_i \rangle \mathbf{w}_i^T \mathbf{u}_j \\ &= \sum_{i=1}^q \sum_{j=1}^p \frac{\lambda_{W_i}}{\lambda_{U_j}} (\mathbf{u}_j^T, \mathbf{w}_i)^2. \quad \square \end{aligned}$$

Equation (10) has a simple geometric interpretation too. Based on (10), (1) weights each pair of vectors  $(\mathbf{u}_j, \mathbf{w}_i)$  according to their agreement. Those pairs with similar vectors will have a higher weight,  $w_{j,i} = (\mathbf{u}_j^T \mathbf{w}_i)^2$ , than those that differ (i.e., those that are orthonormal to each other). Then, for

those pairs that agree (i.e., those vectors of  $\mathbf{M}_U$  and  $\mathbf{M}_W$  that favor a common direction), the goal is to maximize the ratio

$$\frac{\lambda_{W_i}}{\lambda_{U_j}},$$

which is equivalent to (2).

The results presented above have the following interpretation: When the agreeing pair of vectors  $(\mathbf{u}_j, \mathbf{w}_i)$  correspond to a  $\mathbf{u}_j$  with associated small eigenvalue and a  $\mathbf{w}_i$  with associated large variance, the result of (1) is optimal according to Bayes. However, when the  $(\mathbf{u}_j, \mathbf{w}_i)$  pair agreeing correspond to a  $\mathbf{w}_i$  with smaller eigenvalue than that of  $\mathbf{u}_j$ , the results of (1) are not guaranteed to be optimal because we cannot maximize  $|\mathbf{v}^T \mathbf{M}_W \mathbf{v}|$  and minimize  $|\mathbf{v}^T \mathbf{M}_U \mathbf{v}|$  simultaneously.

For example, the goal of NDA is to maximize the distance between the bordering samples of different classes while minimizing the intraclass distance. However, if a vector  $\mathbf{e}_1$  maximizes the first measure and a different vector  $\mathbf{e}_2$  (orthogonal to  $\mathbf{e}_1$ ) minimizes the second, NDA will become unstable and may not be able to successfully compute the optimal result.

This problem was quantitatively measured by (10). When the agreeing vectors,  $(\mathbf{u}_j, \mathbf{w}_i)$ , are such that  $\mathbf{u}_j$  has a small associated eigenvalue and  $\mathbf{w}_i$  has a large one, the value of (10) is large. And, when  $\mathbf{u}_j$  has a large associated eigenvalue and  $\mathbf{w}_i$  a small one, the value of (10) is small. Unfortunately, we do not know how large or how small (10) needs to be to help us determine when the results may be correct or wrong.

A more useful measure is, however, accounted for within (10). Note that the results of (1) may be incorrect when an eigenvector of  $\mathbf{M}_W$ ,  $\mathbf{w}_i$ , is equal or highly correlated to one of the first  $i$  eigenvectors of  $\mathbf{M}_U$ , because the discriminant power will (in such cases) be very small. Therefore, the goal is to minimize

$$\sum_{i=1}^r \sum_{j=1}^i (\mathbf{u}_j^T \mathbf{w}_i)^2,$$

which is the result we gave in Theorem 1, i.e., (8). Our equation can then be used to determine where the results of (1) may not be correct. This is, when the value of  $K$  is large, the result of (1) needs not minimize the Bayes error for the given distributions. Even when the data is homoscedastic and linearly separable.

In general,  $r$  needs to be smaller than or equal to  $\lfloor \frac{q}{2} \rfloor$ . Otherwise, an eigenvector of  $\mathbf{M}_W$  that is equal to one of  $\mathbf{M}_U$  can (almost always) be found and then  $K > 0$  although, in this case, the results may very well be correct. An obvious alternative is to select  $r$  so that  $\lambda_{W_r} / \lambda_{U_r}$  is small compared to  $\lambda_{W_1} / \lambda_{U_1}$ .

Another, arguably more convenient way, to calculate where (1) is applicable, is as follows:

**Theorem 3.** *If for some  $(i, j)$ , with  $i \leq r$  and  $j \leq i$ ,*

$$\max_{i,j} (\mathbf{u}_j^T \mathbf{w}_i)^2 \approx 1, \quad (12)$$

*the basis vectors given by (1) may not minimize the Bayes error of the given data distributions.*

Equation (12) only searches for those vectors that result in a one to one conflict between  $\mathbf{M}_W$  and  $\mathbf{M}_U$ , which in most circumstances is the triggering factor of the problem described above.

Moreover, it is important to note that the results of (1) may still be correct when (12) holds or when  $K > 0$ . This is illustrated in Fig. 3. In this figure, we apply LDA to two three-class examples. The first is the same as that shown in Fig. 2b, which did not produce the correct results. For comparison, we now show a second example in Fig. 3b where we also have  $\mathbf{u}_1 = \mathbf{w}_1$  (same as above). Yet, the result obtained by (1) in this second example is correct. This is the reason why we refer to the cases with  $K > 0$  as examples where (1) is *not stable*, because, in such cases, it is not known a priori whether the results given by (1) will be correct or not. Ideally, we would only want to use (1) where we know it to be stable. But, in practice, we may decide to proceed with caution where the algorithm is known to be unstable.

### 3 ROBUST ALGORITHMS

A common problem with (1) is that of computing the inverse of  $\mathbf{M}_U$ . In many applications in computer vision (e.g., feature extraction), the number of samples is usually much smaller than the number of dimensions and, therefore,  $\mathbf{M}_U$  is singular. In this section, we show derivations to solve this problem. Our solution is based on the geometrical interpretation of (1) given above.

We first note that the subspace spanned by the basis vectors of (1) is, in general, the same as or a subspace of that spanned by the eigenvectors of  $\mathbf{M}_W \mathbf{W} = \mathbf{W} \Lambda_W$  and  $\mathbf{M}_U \mathbf{U} = \mathbf{U} \Lambda_U$ . Note that both matrices,  $\mathbf{M}_W$  and  $\mathbf{M}_U$ , are computed using the same set of samples and, therefore, their basis vectors will (almost always) span a subspace of the PCA space. In those rare cases where this is not so, we can easily solve the problem by adding synthetic samples of such statistics to our matrices.

It is therefore possible to compute the basis vectors of (1) directly from  $\mathbf{W}$  and  $\mathbf{U}$  as follows:

**Theorem 4.** *The basis vectors of (1) are equivalent to those of*

$$\left( \sum_{j=1}^q \hat{\mathbf{w}}_j \mathbf{w}_j^T \right) \mathbf{V} = \mathbf{V} \Lambda,$$

where  $\hat{\mathbf{w}}_j = \sum_{i=1}^p \frac{\lambda_{W_i}}{\lambda_{U_i}} \mathbf{u}_i^T \mathbf{w}_j \mathbf{u}_i$  is the reconstruction of  $\mathbf{w}_j$  with regard to all nonzero eigenvectors of  $\mathbf{M}_U \mathbf{U} = \mathbf{U} \Lambda_U$  and properly scaled by  $\frac{\lambda_{W_i}}{\lambda_{U_i}}$ .

**Proof.** Equation (1) can be reworked as follows:

$$\sum_{j=1}^q \lambda_{W_j} \mathbf{w}_j \mathbf{w}_j^T \mathbf{V} = \mathbf{U} \Lambda_U \mathbf{U}^T \mathbf{V} \Lambda$$

$$\left( \begin{pmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_p^T \end{pmatrix} \lambda_{W_1} \mathbf{w}_1 \mathbf{w}_1^T + \cdots + \begin{pmatrix} \mathbf{u}_1^T \\ \vdots \\ \mathbf{u}_p^T \end{pmatrix} \lambda_{W_q} \mathbf{w}_q \mathbf{w}_q^T \right) \mathbf{V} = \Lambda_U \mathbf{U}^T \mathbf{V} \Lambda.$$

If  $\mathbf{M}_W$  is full ranked,  $p = m$  (where  $m \times m$  is the size of  $\mathbf{M}_U$ ), we can multiply both sides by  $\mathbf{U}\Lambda_U^{-1}$ ,

$$\begin{pmatrix} \frac{1}{\lambda_{U_1}} \mathbf{u}_1 & \cdots & \frac{1}{\lambda_{U_p}} \mathbf{u}_p \end{pmatrix} \begin{pmatrix} \sum_{j=1}^q \lambda_{W_j} \langle \mathbf{u}_1, \mathbf{w}_j \rangle \mathbf{w}_j^T \\ \vdots \\ \sum_{j=1}^q \lambda_{W_j} \langle \mathbf{u}_p, \mathbf{w}_j \rangle \mathbf{w}_j^T \end{pmatrix} \mathbf{V} = \mathbf{V}\Lambda$$

$$\sum_{i=1}^p \sum_{j=1}^q \frac{\lambda_{W_j}}{\lambda_{U_i}} \langle \mathbf{u}_i, \mathbf{w}_j \rangle \mathbf{u}_i \mathbf{w}_j^T \mathbf{V} = \mathbf{V}\Lambda. \quad (13)$$

When  $\mathbf{M}_U$  is not full ranked,  $p < m$ ,  $\sum_{i=1}^m \frac{\lambda_{B_i}}{\lambda_{X_i}} \mathbf{u}_i^T \mathbf{w}_j \mathbf{u}_i = \sum_{i=1}^p \frac{\lambda_{B_i}}{\lambda_{X_i}} \mathbf{u}_i^T \mathbf{w}_j \mathbf{u}_i$ . This is easy to show, since all those  $\mathbf{u}_i$  corresponding to a dimension of zero variance are orthogonal to all  $\mathbf{w}_j$  and, therefore,  $\mathbf{u}_i^T \mathbf{w}_j = 0$ .  $\square$

We find the above result very useful for computing the subspace defined by the first several basis vectors of (1) since it does not require the calculation of the inverse of  $\mathbf{M}_U$ . This is achieved by means of the factorization of (1) derived in (13). This property is especially useful when  $\mathbf{M}_U$  is singular, but can be used wherever one does not want to compute the inverse of  $\mathbf{M}_U$ . For example, this is useful for large matrices, because it saves us of extensive computations. The algorithm of Theorem 4 thus defines a robust, easy way of finding the discriminant features of the data.

In recent years, there has been an intense debate on the use of LDA within the appearance-based approach to object recognition that Theorem 4 resolves. The technique and controversy go as follows: When singularity prevents us to directly use LDA in the original feature space, we can first project the data of our original (high-dimensional) space to that defined by PCA [2], [7], [30], [23]. The goal of this step is to reduce the size of the original space to one where LDA is applicable (i.e., where  $\mathbf{S}_W$  or  $\Sigma_X$  is not singular). Then, we can extract the LDA space from the resulting PCA-space.

This technique has received criticism, because the first reduction, which is carried out by PCA, may eliminate some of the most discriminant features needed to successfully discriminate between several classes. Theorem 4 shows this is not true. Given the training data available, the best we can do is to compute the LDA space from a “normalized” PCA subspace. Our result formally justifies the experimental observation which shows that the LDA/PCA algorithm works well when the data is linearly separable. The theorem above shows that we can get identical results to those we would obtain with LDA computed in the original space using (13). This is not to say that LDA will successfully discriminate among all classes. Rather, that the results obtained using (13) are the same as those carved out by LDA from the original space. To know whether LDA produces a subspace that is able to separate the sample distributions of a set of linearly separable classes, we need to use (8) or (12) defined above.

It is also interesting to note that we can now estimate the discriminant power lost by the algorithms defined in Theorem 4 quite easily.

**Corollary 5.** *Compute the discriminant space using the first  $e$  eigenvectors of  $\mathbf{M}_U$ , i.e.,*

$$\left( \sum_{j=1}^q \tilde{\mathbf{w}}_j \mathbf{w}_j^T \right) \mathbf{V} = \mathbf{V}\Lambda,$$

where now  $\tilde{\mathbf{w}}_j = \sum_{i=1}^e \frac{\lambda_{W_j}}{\lambda_{U_i}} \mathbf{u}_i^T \mathbf{w}_j \mathbf{u}_i$  is the reconstruction obtained with the  $e$  eigenvectors associated to the  $e$  largest eigenvalues of  $\mathbf{M}_U \mathbf{U} = \mathbf{U}\Lambda_U$  and  $e < p$ . Then, it is possible to estimate the discriminant power lost as

$$\sum_{i=1}^q \sum_{j=e+1}^p \frac{\lambda_{W_i}}{\lambda_{U_j}} \left( \mathbf{u}_j^T \mathbf{w}_i \right)^2. \quad (14)$$

Similarly, we can also redefine the result of Theorem 1 for those cases where only the first  $e$  eigenvectors of  $\mathbf{M}_U$  are used. In such a case, the value of  $K$  becomes,

$$\sum_{i=1}^r \sum_{j=1}^{\min(i,e)} \left( \mathbf{u}_j^T \mathbf{w}_i \right)^2. \quad (15)$$

Before we conclude this section, it is also interesting to mention that one usually prefers to compute LDA using the sample covariance matrix rather than the within-class scatter matrix, i.e.,  $\mathbf{M}_U = \Sigma_X$ . The reasons for that are as follows: First, because, in general, the covariance matrix is of higher rank than the within-class scatter matrix, which allows us to use LDA in spaces of higher dimensionality. This means that  $\Sigma_X$  may contain additional information of the data that is not available in  $\mathbf{S}_W$ . And, second, because if  $\mathbf{S}_W$  and  $\Sigma_X$  are nonsingular matrices, the results obtained with either matrix only differ in the magnitude of their eigenvalues. This last result can be formally stated as follows [10]:

**Remark 6.** If  $\mathbf{S}_W$  and  $\Sigma_X$  are full rank matrices, then the basis vectors of LDA as defined by  $\mathbf{M}_W = \mathbf{S}_B$  and  $\mathbf{M}_U = \mathbf{S}_W$  are the same as those obtained with  $\mathbf{M}_W = \mathbf{S}_B$  and  $\mathbf{M}_U = \Sigma_X$  and those obtained with  $\mathbf{M}_W = \Sigma_X$  and  $\mathbf{M}_U = \mathbf{S}_W$ , except for a possible difference in the scaling of their eigenvalues.

The use of  $\Sigma_X$  does carry an important disadvantage though. Since  $\Sigma_X = \mathbf{S}_B + \mathbf{S}_W$  (i.e.,  $\mathbf{S}_B$  is part of the definition of  $\mathbf{M}_W$  and  $\mathbf{M}_U$ ), the result of (1) is expected to be less stable when one uses  $\Sigma_X$  than when one uses  $\mathbf{S}_W$ .

To conclude this section, we would like to point out the similarities and differences between the robust algorithm defined in Theorem 4 above and those defined in the past.

We have first noted that when  $\mathbf{M}_U = \Sigma_X$ , our algorithm parallels those that use PCA as a first step to avoid singularity problems. These results are, nonetheless, extendable to other metrics. Remark 6 shows how our results extend to the case of  $\mathbf{M}_U = \mathbf{S}_W$  and, in general, scale variations also apply to other metrics.

We should also point out, however, that when  $\mathbf{M}_U = \mathbf{S}_W$  but the rank of  $\mathbf{S}_W$  is smaller than that of  $\Sigma_X$  (and, therefore, the range of  $\mathbf{S}_W$  is a subspace of  $\text{ran}(\Sigma_X)$ ), the results given by either (1) or (13) (which are identical) need not minimize the Bayes error. In this case, the discriminant information lost can be estimated using (14), where the second summing term now represents the basis vectors of the null space of  $\mathbf{S}_W$ , i.e., the difference between the  $\text{ran}(\Sigma_X)$  and the  $\text{ran}(\mathbf{S}_W)$ .

In other circumstances, the rank of  $\mathbf{M}_W$  is smaller than that of  $\mathbf{M}_U$ . In such cases, the space spanned by  $\mathbf{M}_W$  is a subspace of that spanned by  $\mathbf{M}_U$  and, therefore, we may find it more appropriate to solve (1) on the space of  $\mathbf{M}_W$  [10], [35]. This means, we will first compute those eigenvectors of  $\mathbf{M}_W$ , which are given by

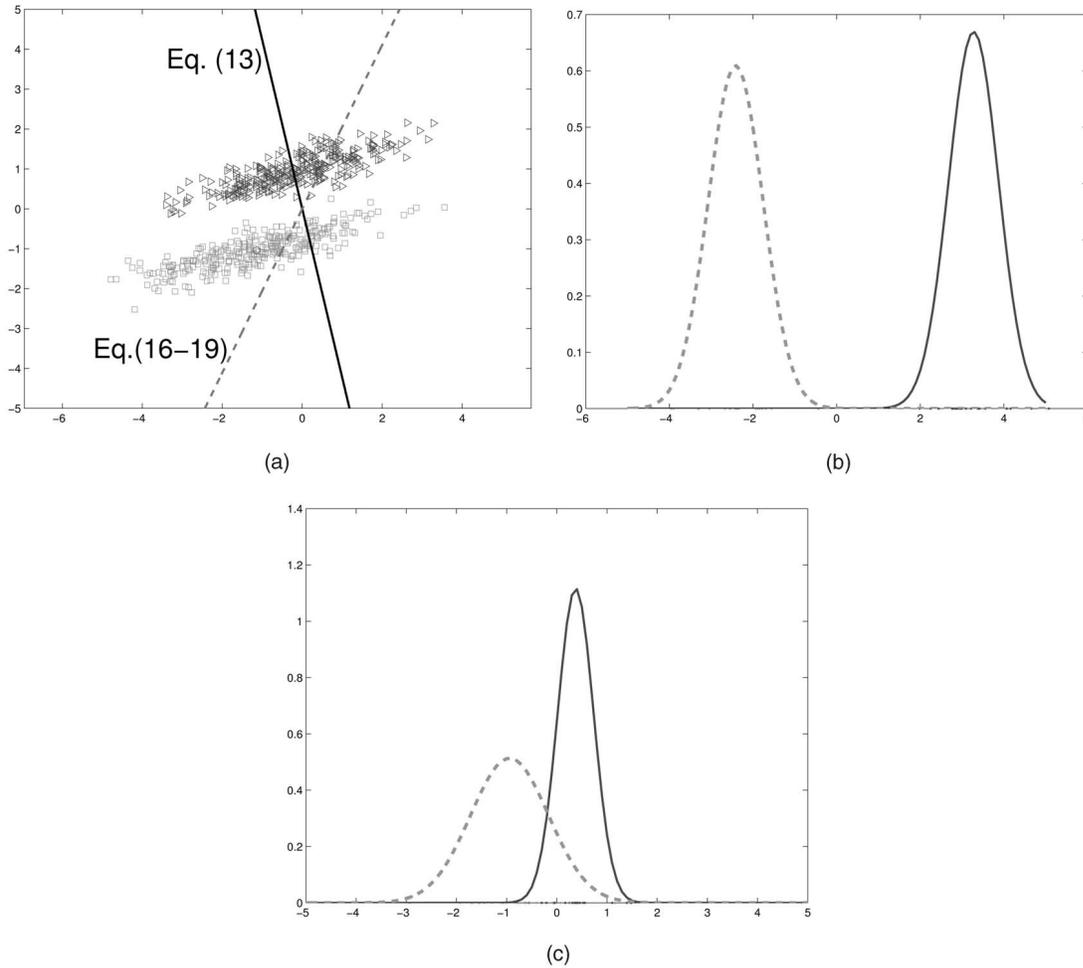


Fig. 4. (a) In this two-dimensional example, the result obtained by the robust method defined in Theorem 4 is correct, whereas those computed with other approaches are not. Note that, in this example,  $M_U$ , which was made equal to the covariance matrix of the data, is not singular. In (b) and (c), we have two Gaussian distributions embedded in a high-dimensional space, where now  $M_U$  is singular. (b) Shows the results of our robust algorithm defined in Theorem 4. (c) Shows the result obtained using (16), (17), (18), and (19). The result shown in (b) minimizes the Bayes error, but that of (c) does not.

$$M_W W = W \Lambda_W \quad (16)$$

and then recover the eigenvectors  $\mathbf{V}$  of (1) by projecting  $M_U$  onto the space of  $M_W$ ,

$$\begin{aligned} \mathbf{Z} &= \mathbf{W} \Lambda_W^{-1/2}, \\ \mathbf{Y} &= \mathbf{Z}^T M_U \mathbf{Z}, \end{aligned} \quad (17)$$

calculate the basis vectors in the reduced space,

$$\mathbf{Y} \mathbf{V}_Y = \mathbf{V}_Y \Lambda_Y, \quad (18)$$

and, finally, map these results back to our original feature space,

$$\mathbf{V} = \Lambda_Y^{-1/2} \mathbf{V}_Y^T \mathbf{Z}^T. \quad (19)$$

This algorithm is adequate in some cases and, generally, yields reasonable results. However, cases exist where the algorithm defined in (16), (17), (18), and (19) does not produce the correct solution. In Fig. 4, we show two examples for the LDA algorithm with  $M_W = \mathbf{S}_B$  and  $M_U = \Sigma_X$ . In Fig. 4a, we have a simple two-dimensional example where  $\Sigma_X$  is not singular. In this case, the result obtained with the algorithm defined in (16), (17), (18), and (19) does not correspond to the desirable one. For comparison, we have shown the result we

would obtain using the algorithm described in Theorem 4. Of course, our result is the same we would get with (1), which (in this case) is the correct one.

Figs. 4b and 4c show the projection of two Gaussians embedded in a high-dimensional space (where now  $\Sigma_X$  is singular) onto the one-dimensional space computed by the algorithm of Theorem 4 and that defined in (16), (17), (18), and (19). As we can see in the figure, our algorithm finds the optimal (according to Bayes) solution, while the other method does not.

## 4 CLASSIFICATION

In general terms, an algorithm generalizes when the empirical error (which is obtained using the training samples) is a close match of the test error. In what follows, we assume generalization to mean that the empirical error is a close match of the Bayes error of the underlying data distributions. This is a useful definition within the context of feature extraction as used in this paper.

In this section, we briefly outline some of the generalization problems of (1). We also argue that variations on the value of  $K$  when the  $i$ th sample is left out can be used to determine where the linear methods discussed above may

not generalize well. By understanding when (1) does not generalize well, we are in a position to define better algorithms in the future and, at present, help determine where current algorithms cannot be successfully applied.

Examples demonstrating how and where we can apply distinct linear algorithms were presented in the sections above. These corresponded to examples on synthetic data where we knew the classes were linearly separable. While our results are best and most easily applicable to such cases, we can also use our criteria to understand why some techniques work on some nonlinearly separable problems while others do not. For illustration, we will work with an example using two real data sets of objects for which the Ho-Kashyap test [6] shows the classes are not linearly separable. The first of these sets is the ETH-80 database [19], which contains several images of the following visual categories: apples, pears, cars, cows, horses, dogs, tomatoes, and cups. Each category includes the images of 10 objects (e.g., 10 different pears) photographed at a total of 41 orientations, which gives a total of 410 sample images per category. The second set is the AR-face database [23]. The AR-database consists of frontal-view faces of a large number of subjects. Each person appears photographed under different lighting conditions and distinct facial expressions and some images have partial occlusions. The first 13 images for a total of 100 individuals will be used in our test.

In this example, we use the appearance-based framework, where each image pixel represents a dimension (feature). This means that for the images in the ETH-80 database, we will crop the smallest window that circumscribes the whole object and then resize all the resulting subimages to a standard size of 25 by 30 pixels. This gives a feature space of 750 dimensions. The same procedure is applied to the face images, except that these are reduced to a common size of 21 by 29 pixels. Hence, our faces will be represented as vectors in a feature space of 609 dimensions.

The two subspaces defined by  $\mathbf{M}_U = \Sigma_X$  and  $\mathbf{M}_W$  equal to  $\mathcal{S}_B$  (namely, LDA) and  $\mathcal{S}_B$  (NDA) are computed using all but one sample image. The sample left out is then used for testing using the Euclidean-based nearest neighbor algorithm. This is repeated for all possible ways of leaving one sample out. The results obtained with this leave-one-sample-out test can then be analyzed with the use of our criteria. For example, our implementation of LDA (as defined in Theorem 4) performs well for faces (with a successful classification rate of  $\sim 98$  percent), but worse on objects ( $\sim 70$  percent). While the number of  $K$  depend on  $r$  and, thus, cannot be used directly, we can (nonetheless) compare the value of  $K$  once this has been properly normalized by  $r$ . In our example,  $r$  can be made equal to the number of classes minus 1,  $r = c - 1$ , since this is smaller than  $q$ . We see that while for faces  $K/r \approx 0.2$ , for objects  $K/r \approx 0.34$ . The probability of misclassification (as given by Theorem 1) in the category problem is higher and, therefore, our implementation should only be used with care. We also note that our implementation of NDA gives  $K/r \approx 0.68$  when applied to objects. This indicates that NDA will most probably perform poorly on the ETH-80 data set. This result is confirmed by our leave-one-sample-out test, where the successful recognition rate is about 33 percent.

In the example above, we could have also used the result of Theorem 3. Similarly to what we did above, we can propose a normalized version of that result,

$$\tilde{K} = \frac{1}{r} \sum_{i=1}^r \max_{\forall j \leq i} (\mathbf{u}_j^T \mathbf{w}_i)^2.$$

Using this equation, we see that  $\tilde{K} \approx 0.06$ , a very low value, when we apply LDA to our data set of face images. This means that LDA should work well on faces—result confirmed by our leave-one-out test with a recognition rate of about 98 percent. However,  $\tilde{K} \approx 0.32$  when LDA is used on the categorization problem, which means the classification results will now be considerably lower. This is once more ratified by our test, where the recognition rate is only about 70 percent. Similarly,  $\tilde{K} \approx 0.6$  when NDA is applied to the object categorization problem, and  $\tilde{K} \approx 0.3$  when applied to faces. This indicates that the probability of misclassification (or, in general terms, of poor generalization) on the category task is much higher than that of faces. These results are consistent with the leave-one-sample-out test, where faces are successfully classified about 92.4 percent of the time and objects only about 33 percent.

One may still wonder how much of the low performance of LDA and NDA on the ETH-80 data set is due to the fact that the data is not linearly separable and how much to the fact that our criteria predicted that these would not perform adequately on this set. To further study this, we have computed the classification accuracy of a linear Support Vector Machine (SVM) with soft margin and that of a K-SVM (Kernel-SVM) using the same leave-one-object-out test. In this experiment, we see that while the successful classification rate for the K-SVM (with a Gaussian kernel) is above 86 percent that of the soft-margin SVM is only about 75 percent (which may be interpreted as an approximation to an upper bound of the classification accuracy of a linear method). These results confirm our two main findings. First that the data is not linearly separable (since K-SVM is able to outperform SVM [34]). And, second, that the values of  $K/r$  and  $\tilde{K}$  are good predictors of where a linear feature extraction will work, even when the data is not linearly separable.

From the results discussed in the preceding paragraphs, we first note that the value of  $K$  should, at most, be proportional to the number of eigenvectors of  $\mathbf{M}_W$  and  $\mathbf{M}_U$ . To see this, we need to go back to the definition of  $K$  given in Theorem 1, (8). There, the  $i$ th eigenvector of  $\mathbf{M}_W$  can only be in conflict with the first  $i$  eigenvectors of  $\mathbf{M}_U$ . Therefore, an upper bound of  $K$  is given by the total number of possible combinations. And, although the result of (1) is not guaranteed to be correct if  $K \neq 0$ , in practice  $K$  can be allowed to grow when the number of eigenvectors of  $\mathbf{M}_W$ ,  $q$ , increases. Of course, this does not guarantee success, but may be used as a measure of generalization. Unfortunately, when each eigenvector of  $\mathbf{M}_W$  is in conflict with one of  $\mathbf{M}_U$ , the results will be incorrect most of the time. This last observation can be summarized as follows: Define

$$a_i = \max_{j \leq i} (\mathbf{u}_j^T \mathbf{w}_i)^2. \quad (20)$$

If  $a_i \approx 1$  for all  $i \leq r$ , the results of (1) will generally not minimize the Bayes error of the actual data distributions. That is to say, the results will not necessarily generalize.

This means that, in practice, if  $K/r$  remains low (as  $r$  either decreases or increases), the results given by (1) can be expected to generalize well. When  $K/r$  is high, the results do not usually generalize. In such cases, one may want to turn to nonlinear methods or, if the goal is classification only, to other types of linear algorithms such as SVM [32], [6], although this comes to a considerably extra computational cost.

A somehow different way of estimating where a learning algorithm generalizes is to look at its stability. One possibility, which has recently proven success, is to study what happens when one of the training samples is left out [4], [26]. Typically, if the results (or the empirical error) do not change when different samples are left out, the algorithm is said to be stable. Under some conditions, such stabilities imply generalization.

In our case, we would like to know whether (8) or (20) remain close to zero (or, alternatively, proportional to  $r$ ) when different training samples are left out. If this happens and the sample metrics are close matches of the real ones (even if one of the samples is left out), we can assume the algorithm generalizes well. In practice, stability and generalization may be given when  $K_i \approx K_j$  for all  $i \neq j$  and  $K_i < r$  or when most  $a_i$  are approximately zero; where  $K_i$  is the value of (8) when the  $i$ th sample is left out.

The criteria discussed in the preceding paragraphs are important, mainly, because they can be readily used to improve learning or develop new classification and feature extraction algorithms. For example, we may use the results shown above to select which samples are most adequate for training. In this case, those samples which do not preserve stability would be left out. Alternatively, we can use our results to perform online modifications of the metrics to be maximized or minimized. Gradient ascent (descent) techniques would be appropriate for this. Or, we could define a linear combination of known metrics and use those regularizing parameters which maximize stability. Finally, we could also tackle this problem by restricting the size of the hypothesis space on the bases of stability. The hypothesis space is that which contains all possible functions (in our case, metrics) from which the learning algorithm searches a solution. Of course, much research is still needed before we can develop and test these (robust) linear algorithms, but the results reported in this paper are an important milestone toward this goal.

Another important step toward the design of classifiers and feature extraction algorithms robust to the problems defined in this paper, is to define accurate measures of their discriminant power. Note that the discriminant measure of (1), which is given by (9), gives a pessimistic estimate, because it only considers those vectors that are in conflict (or agreement, if one considers  $U$  to be the eigenvectors of  $M_U^{-1}$ ) and ignores those that are orthogonal to each other. In other words, if a conflict is detected, the method assumes the algorithm will not be able to classify the data correctly. However, and as previously illustrated in Fig. 3b, this is not necessarily so. Several new measures can be proposed based on the results presented in this paper. And, new classification algorithms can be proposed as gradient descent techniques based on any new discriminant equation. The results of Theorem 1 can be used to give sufficient conditions for (1) to work. And, hence, our results can also be used to study or improve current techniques; as it is the case, for example, of [33], [8], [29], [21], [18], [14].

Finally, we would like to bring a point of caution to the results discussed in this section. While, above, some criteria were cited as sufficient conditions for generalization, these usually assume that a sufficiently large number of training samples representing the shape of the data distributions is available. Caution needs to be taken when the number of samples is small, especially when the number of samples is smaller than the number of dimensions. When the value of  $K$  remains low and approximately constant but the number of samples is small, (1) usually results in overfitting. This means that the training data is very well learned (and usually optimally separated), but that the results *do not generalize* to new samples. The main reason for such result is given by the known fact that when the number of samples is much smaller than the number of dimensions, the classes can be generally separated with many distinct hyperplanes. While this problem has been largely studied within some contexts [32], it remains largely unsolved in multivariate analysis and, in particular, in feature extraction.

The problem described in this paper is not only encountered in computer vision problems [23], [3], [15]. In other areas, such as bioinformatics, this is also predominant. For example, in the classification of microarray data [27], where the number of samples (patients) is generally much smaller than the number of dimensions (genes). Research in this area has led to results that are rarely reproducible. While the technique has been usually cited as the most probable cause of this failure, the problem is typically due to overfitting not algorithmical.

## 5 CONCLUSIONS

In this paper, we have shown that linear feature extraction algorithms are not always guaranteed to minimize the Bayes error even if the metrics used ( $M_W$  and  $M_U$ ) do so in principal. We have shown that the sanity check is not that of whether the data is homoscedastic or heteroscedastic (which had been our common practice). Instead, we have proven that the results are not guaranteed to be correct when the smallest angle between the  $i$ th eigenvector of  $M_W$  and the first  $i$  eigenvectors of  $M_U$  is close to zero. We have proposed two distinct ways to account for this. These are summarized in Theorems 1 and 3.

We have then used our results to define a new robust algorithm (of such linear methods) in Theorem 4. And, we have discussed how these results can be used to improve the design of current algorithms. Extensions of our results to define more robust learning algorithms were also given. In our discussion, we made emphasis to the discriminant power obtained with such linear methods (as, for example, that of different types of discriminant analysis). Here, we have shown that the classical discriminant power of the linear feature extraction algorithms defined in this paper is not adequate (Theorem 2) and that alternatives need to be found.

We believe our results will be useful to those working on feature extraction algorithms and those that need to apply them to their problems. We would like to point out, however, that the linear methods defined in (1) are not only typical of linear feature extraction algorithms. A review of any computer vision book will reveal that such equations are common place in our community. The results reported in this paper will help scientist in computer vision to better understand their algorithms and, most importantly, improve upon

them. Ideally, and according to a well-known classical view (Occam's razor), we want to use the simplest possible technique known to work properly. Understanding the limits of current approaches is essential to this task.

Obviously, feature extraction techniques are applicable to a large variety of problems in science and, therefore, one can expect that the result reported in this paper will also impact areas of research as diverse as genetics, economics, climate modeling, and neuroscience.

## APPENDIX A

### NOTATION

$\mathbf{M}_W$  is a symmetric positive defined matrix that defines the metric to be maximized.

$q$  is the rank of  $\mathbf{M}_W$ .

$\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_q\}$  are the eigenvectors of  $\mathbf{M}_W$ .

$\Lambda_W = \{\lambda_{W_1}, \dots, \lambda_{W_q}\}$  the corresponding eigenvalues.

$\mathbf{M}_U$  defines the metric to be minimized.

$p$  is the rank of  $\mathbf{M}_U$ .

$\mathbf{U} = \{\mathbf{u}_1, \dots, \mathbf{u}_p\}$  are the eigenvectors of  $\mathbf{M}_U$ .

$\Lambda_U = \{\lambda_{U_1}, \dots, \lambda_{U_p}\}$  the corresponding eigenvalues.

$\mathbf{S}_B$  is the sample between-class scatter matrix.

$\mathbf{S}_B$  is the sample nonparametric between-class scatter matrix.

$\mathbf{S}_W$  is the sample within-class scatter matrix.

$\Sigma_\rho$  is a matrix that defines the higher moments of the data.

$\Sigma_X$  is the sample covariance matrix of the data.

### ACKNOWLEDGMENTS

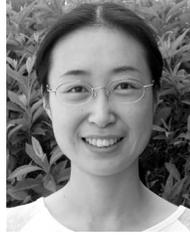
The authors would like to thank the referees for their insightful comments. They would also like to thank Mar Jimenez for useful discussion. This research was supported in part by the US National Institutes of Health under grant R01 DC 005241.

### REFERENCES

- [1] M.S. Bartlett, "Face Image Analysis by Unsupervised Learning," *Kluwer Int'l Series on Eng. and Computer Science*, vol. 612, 2001.
- [2] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, July 1997.
- [3] R. Beveridge, K. She, B. Draper, and G. Givens, "A Nonparametric Statistical Comparison of Principal Component and Linear Discriminant Subspaces for Face Recognition," *Proc. Computer Vision and Pattern Recognition*, pp. 1:535-542, 2001.
- [4] O. Bousquet and A. Elisseeff, "Stability and Generalization," *J. Machine Learning Research*, vol. 2, pp. 499-526, 2002.
- [5] R.D. Cook, *Regression Graphics*. Wiley, 1998.
- [6] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, second ed. Wiley, 2001.
- [7] K. Etemad and R. Chellapa, "Discriminant Analysis for Recognition of Human Face Images," *J. Optics of Am. A*, vol. 14, no. 8, pp. 1724-1733, 1997.
- [8] M.A.T. Figueiredo and A. Jain, "Unsupervised Learning of Finite Mixture Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381-396, Mar. 2002.
- [9] K. Fukunaga and J.M. Mantock, "Nonparametric Discriminant Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 5, pp. 671-678, 1983.
- [10] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, second ed. Academic Press, 1990.
- [11] G.H. Golub and C.F. van Loan, *Matrix Computations*, third ed. John Hopkins Univ. Press, 1996.
- [12] H. Gupta, A.K. Agrawal, T. Pruthi, C. Shekhar, and R. Chellappa, *An Experimental Evaluation of Linear and Kernel-Based Methods for Face Recognition*. John Hopkins Univ. Press, 1996.
- [13] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer-Verlag, 2001.
- [14] X. He, S. Yan, Y. Hu, P. Niyogi, and H.J. Zhang, "Face Recognition Using Laplacian Faces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328-340, Mar. 2005.
- [15] A.K. Jain and B. Chandrasekaran, "Dimensionality and Sample Size Considerations in Pattern Recognition Practice," *Handbook of Statistics*, P.R. Krishnaiah and L.N. Kanal, eds., vol. 2, pp. 835-855, 1982.
- [16] A.K. Jain, R.P.W. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4-37, Jan. 2000.
- [17] I.T. Jolliffe, *Principal Component Analysis*, second ed. Springer-Verlag, 2002.
- [18] T.K. Kim and J. Kittler, "Locally Linear Discriminant Analysis for Multimodally Distributed Classes for Face Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 318-327, Mar. 2005.
- [19] B. Leibe and B. Schiele, "Analyzing Appearance and Contour Based Methods for Object Categorization," *Proc. IEEE Computer Vision and Pattern Recognition*, June 2003.
- [20] J. Li, "Sliced Inverse Regression for Dimensionality Reduction," *J. Am. Statistical Soc.*, vol. 86, no. 414, pp. 316-327, 1991.
- [21] M. Loog and R.P.W. Duin, "Linear Dimensionality Reduction via a Heteroscedastic Extension of LDA: The Chernoff Criterion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 6, pp. 732-739, June 2004.
- [22] K.V. Mardia, J.T. Kent, and J.M. Bibby, *Multivariate Analysis*. Academic Press, 1979.
- [23] A.M. Martinez and A.C. Kak, "PCA versus LDA," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228-233, Feb. 2001.
- [24] G.J. McLachlan, *Discriminant Analysis and Statistical Pattern Recognition*. Wiley-Interscience, 1992.
- [25] K.E. Muller, "Understanding Canonical Correlation through the General Linear Model and Principal Components," *Am. Statistician*, vol. 36, pp. 342-354, 1982.
- [26] T. Poggio, R. Rifkin, S. Mukherjee, and P. Niyogi, "General Conditions for Predictivity in Learning Theory," *Nature*, vol. 428, pp. 419-422, 2004.
- [27] D.F. Ransohoff, "Opinion—Rules of Evidence for Cancer Molecular-Marker Discovery and Validation," *Nature Rev. Cancer*, vol. 4, pp. 309-314, 2004.
- [28] C.R. Rao, *Linear Statistical Inference and Its Applications*, second ed. Wiley Interscience, 2002.
- [29] I. Robledo and S. Sarkar, "Statistical Motion Model Based on the Change of Feature Relationships: Human Gait-Based Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 10, pp. 1323-1328, Oct. 2003.
- [30] D.L. Swets and J.J. Weng, "Using Discriminant Eigenfeatures for Image Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831-836, 1996.
- [31] W.C. Thacker, "Metric-Based Principal Components: Data Uncertainties," *Tellus*, vol. 48, no. A, pp. 584-592, 1996.
- [32] V.N. Vapnik, *The Nature of Statistical Learning Theory*, second ed. Springer-Verlag, 2000.
- [33] N. Vaswani and R. Chellappa, "Classification Probability Analysis of Principal Component Null Space Analysis," *Proc. Int'l Conf. Pattern Recognition*, 2004.
- [34] M.-S. Yang, "Kernel Eigenfaces vs. Kernel Fisherfaces: Face Recognition Using Kernel Methods," *Proc. Fifth Int'l Conf. Automatic Face and Gesture Recognition*, pp. 215-220, 2002.
- [35] H. Yu and J. Yang, "A Direct LDA Algorithm for High-Dimensional Data—with Applications to Face Recognition," *Pattern Recognition*, vol. 34, pp. 2067-2070, 2001.
- [36] M. Zhu and A.M. Martínez, "Optimal Subclass Discovery for Discriminant Analysis," *Proc. IEEE Workshop Learning in Computer Vision and Pattern Recognition*, 2004.



**Aleix M. Martínez** is an assistant professor in the Department of Electrical and Computer Engineering at The Ohio State University (OSU). He is also affiliated with the Department of Biomedical Engineering and to the Center for Cognitive Science, both at OSU. Prior to joining OSU, he was affiliated with the Electrical and Computer Engineering Department at Purdue University and with the Sony Computer Science Lab. He coorganized the First IEEE Workshop on Computer Vision and Pattern Recognition for Human-Computer Interaction (2003) and served as a cochair for the Second IEEE Workshop on Face Processing in Video (2005). In 2003, he served as a coguest editor of the special issue on face recognition in the journal *Computer Vision and Image Understanding*. He is a coauthor of the AR-face database and the Purdue ASL database. His areas of interest are learning, vision, linguistics, and their interaction. He is a member of the IEEE.



**Manli Zhu** received the MS degree in electrical engineering in 2002 from Zhejiang University, China. She is currently a PhD candidate in the Department of Electrical and Computer Engineering at The Ohio State University. Her research interests are in the field of statistical pattern recognition, especially in feature extraction. She is also working on object classification and face recognition.

▷ **For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).**