Multi-class Pairwise Linear Dimensionality Reduction Using Heteroscedastic Schemes

Luis Rueda[*], B. John Oommen[†] and Claudio Henríquez[‡]

Abstract

Linear Dimensionality Reduction (LDR) techniques have been increasingly important in Pattern Recognition (PR) due to the fact that they permit a relatively simple mapping of the problem onto a lower-dimensional subspace, leading to simple and computationally efficient classification strategies. Although the field has been well developed for the two-class problem, the corresponding issues encountered when dealing with multiple classes are far from trivial. In this paper, we argue that, as opposed to the traditional LDR multi-class schemes, if we are dealing with multiple classes, it is not expedient to treat it as a multi-class problem *per se*. Rather, we shall show that it is better to treat it as an ensemble of Chernoff-based two-class reductions onto different subspaces, whence the overall solution is achieved by resorting to either *Voting*, *Weighting*, or to a *Decision Tree* strategy. The experimental results obtained on benchmark datasets demonstrate that the proposed methods are not only efficient, but that they also yield accuracies comparable to that obtained by the optimal Bayes classifier.

Keywords: *Linear Dimensionality Reduction, Fisher's Discriminant Analysis, Heteroscedastic Discriminant Analysis, Chernoff-based Dimensionality Reduction, Pairwise Multiclass Classification.*

[*]Senior Member, IEEE. School of Computer Science, University of Windsor, 401 Sunset Ave., Windsor, ON, N9B 3P4, Canada. Phone: +1 253-3000 x 3002. E-mail: lrueda@uwindsor.ca. The work of this author was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada, Grant No. RGPIN 261360.

[†]*Chancellor's Professor*; *Fellow : IEEE* and *Fellow : IAPR*. Address : School of Computer Science, Carleton University, 1125 Colonel By Dr., Ottawa, ON, K1S 5B6, Canada. This author is also an *Adjunct Professor* with the University of Agder in Grimstad, Norway. E-mail: oommen@scs.carleton.ca. The work of this author was partially supported by NSERC, the Natural Sciences and Engineering Research Council of Canada.

[‡]Department of Computer Science, University of Concepción, Edmundo Larenas 215, Concepción, 4070409, Chile. The work of this author was partially supported by Chilean National Fund for Scientific and Technological Development, FONDECYT grant No. 1060904. E-mail: chenriquezv@udec.cl

1 Introduction

One of the most ironic situations that arises in the field of statistical Pattern Recognition (PR) is the so-called "curse of dimensionality". The irony, which researchers and practitioners have had to wrestle with even from the infancy of research in this field, can be informally presented as follows:

> *If the patterns to be recognized are represented in a feature space of small dimensions, it is likely that many crucial discriminating characteristics of the classes are ignored. However, if on the other hand, the dimensions of the feature space are large, we encounter this "curse", which brings along the excess baggage of all the related problems associated with learning, training, representation, computation and classification [3, 7, 10].*

The "dimensionality reduction" problem involves reducing the dimension of the input patterns and yields the following advantages [28]:

- We need to extract and retain only the efficient features that yield superior classification in the reduced subspace, which could provide a reliable classification with these limited "patterns";
- We can remove redundant information from the patterns, which, in turn, leads to reduced storage and computation.
- We can project the data onto a lower-dimensional space, (hopefully, a space that can be visualized), which will help us better discern and take advantage of the data distribution and separability.

The literature reports numerous strategies that have been used to tackle this problem. The most well-known of these is Principal Components Analysis (PCA) (the details of which are omitted here) to compute the basis (eigen) vectors by which the data subspaces are spanned, thus retaining the most significant aspects of the structure in the data [7, 10, 28]. Indeed, it should also be clarified that, in this context, the aspects of the structure that are significant are only those which have to do with the variances; features that are significant in terms of discrimination may have low variance, and may therefore be lost altogether. The basic idea of PCA is to represent d-dimensional data by a set of orthogonal directions - capturing most of the variances in the data. While PCA finds components that are efficient for *representation*, the class of Linear Dimensionality Reduction (LDR) strategies seek features that are efficient for *discrimination* [7, 10]. LDR methods attempt to

effectively use the concepts of the within-class scatter distributions, and the between-class scatter distributions, to, hopefully, maximize the separation criterion, as it will be explained presently.

1.1 Rationale for this Paper

As argued above, and in the literature, LDR schemes are effective techniques in PR as they allow us to deal with high dimensional classification problems in a simple and convenient way. These kinds of schemes can be easily implemented for two classes as they involve a simple linear algebraic transformation carried out as a matrix multiplication. When dealing with more than two classes (which is the primary focus of this paper), however, the linear reduction can be solved by an extension of the two-class scheme invoking a *single* linear algebraic operation. Here, again, we encounter the "curse" of dimensionality: Well-separated classes in a higher dimensional space can substantially overlap in a lower dimensional space after performing a single linear transformation. Thus, the handicaps of methods which use such a philosophy can be listed as follows:

1. Multi-class problems are intrinsically different from two-class problems. This is because, while in the two-class case, a single hyperplane can effectively approximate the optimal discriminant function, in the multi-class case, we need the corresponding hyperplanes for *each pair of classes*. Consequently, the optimal projection of any hyperplane need not necessarily lead to the optimal projection of any of the others.

2. In the two-class case, Fisher's discriminant solution leads to a projection which is at most one-dimensional. As opposed to this, for the case of multi-class problems, Fisher's discriminant solution for c classes can reduce the space to a dimension up to $c-1$. Observe that the process of using a *single* linear transformation to yield a one-dimensional projection does not explicitly utilize the latter property advantageously when it seeks to extend the result to a multidimensional multi-class scenario.

3. Our conjecture is that a common projection matrix for the data points of all the classes can be perceived to be a *mixture* of $\binom{c}{2}$ projection matrices. Thus, our hypothesis is that by investigating the classes in a pairwise manner, we are effectively decomposing the overall projection matrix into its composite mixture components, which provide information about the separability of the classes, and such information is missed by merely processing the mixture.

4. There are applications in which the speed of the classification phase is critical. One example of this application is the e-mail spam detector or filter, which typically processes thousands, if not millions, of e-mails per second. In this application, e-mails should be classified very quickly, rather than keeping them on a queue.

5. The final reason for considering the multi-class problem as a set of two-class problems is that it minimizes ambiguous regions, namely, those in which multiple classes overlap.

It is appropriate to mention that a long term goal of this research is to extend these concepts for non-linear classifiers. Indeed, being essentially linear algorithms, neither PCA nor LDA can be of significant relevance to effectively classify data that is inherently nonlinear, which constitutes a primary limitation of linear transformation methods. Consequently, a vast body of research has gone into resolving this limitation, and a detailed review of this is found in [3]. The state-of-the-art in dealing with nonlinear methods include an adaptive method utilizing a rigorous Gaussian distribution assumption [15], the Kernel-based PCA (KPCA) methods [17, 22, 23, 24], the Kernel-based FDA (KFDA), and its reformative variant, the reformative KFDA [29], among others. Thus the exciting problem that deserves attention, and which will hopefully result from this research, is that of designing multi-class Chernoff-bound-oriented solutions to such "non-linear" classifiers, whether they be kernel-based or otherwise.

To satisfactorily respond to these queries, we shall explain how an ensemble of two-class LDR classifiers can be effectively used to solve the multi-class LDR problem: the main goal of this paper.

1.2 Contributions of the Paper

The main contributions of the paper can be sumarized as follows:

1. We show that if we want to design a multi-class LDR scheme, it is expedient for us to design it as an ensemble of $\binom{c}{2}$ two-class LDR schemes, rather than resorting to a single multi-class LDR schemes.

2. Although there are many two-class LDR classifiers, we show that the one which involves a Chernoff-based criterion [20] is the most suitable one.

3. We have undertaken a systematic study of how the results of the individual two-class classifiers can be fused, i.e., by either a *Voting*, *Weighting*, or a *Decision Tree* strategy.

4. The results that have been obtained are quite conclusive, and are based on an extensive testing using benchmark datasets.

1.3 Organization of the Paper

The paper is organized as follows. We first present an informal discussion of the state-of-the-art methods in Section 1.4, while Section 1.5 discusses the most important families of strategies for the two-class scenario. Section 2 outlines the general "all-at-once" multi-class schemes using separability criteria that are extended versions of the corresponding two-class criteria. Section 3 then presents the various pairwise multi-class methods, and introduces the way by which they can be fused. Section 4 describes the experimental results obtained by testing the various methods using the benchmark datasets from the UCI Machine Learning Repository, after which Section 5 concludes the paper.

1.4 Previous LDR Methods

Various schemes that perform LDR have been proposed so far. The most traditional LDR scheme is the well known Fisher's discriminant analysis (FDA) [4], and its many extensions: the *direct* Fisher's discriminant analysis [32], the combined principal component analysis (PCA) and linear discriminant analysis (LDA) [31], and the kernelized PCA and LDA [30]. An improvement to FDA that decomposes classes into subclasses has been proposed in [16]. Also, a scheme to find an optimal kernel over a convex set of kernels has been recently proposed for the kernelized FDA [11].

On the other hand, Rueda and Oommen [21] showed that the optimal classifier between two normally distributed classes can be linear even when the covariance matrices *are not equal*, thus leading to an alternate way of linearly reducing the dimensionality of the space in which the classification is done. A new approach to selecting the *best hyperplane classifier* (BHC) was introduced in [19], which was based on the results related to the optimal pairwise linear classifier. A computationally intensive method for LDR was proposed in [18], which aims to minimize the classification error in the transformed space and operates by computing (or approximating) the *exact* values for the integrals. This approach, though extremely time consuming and prohibitive for high dimensions, does not guarantees an optimal LDR. Another criterion used for dimensionality reduction is the subclass discriminant analysis [34], which aims to optimally divide the classes into subclasses, and then perform a reduction followed by classification.

Of the approaches that have been proposed to generalize homoscedastic-like methods, i.e. FDA, the following represent the state-of-the-art. The first scheme we mention is Heteroscedastic Discriminant Analysis (HDA), proposed in [14]. In that paper, the authors utilize the concept of *directed distance matrices*, and a linear transformation in the original space, to effectively generalize the FDA criterion. They achieve this by substituting the between-class scatter matrix with a weighted sum of the corresponding *directed distance matrices*. Another technique is Chernoff-based Discriminant Analysis (CDA), proposed in [20], where the authors maximize the separability of the lower-dimensional classes measured in terms of the Chernoff distance. In [25, 26], a linear discriminant analysis approach has been proposed, which uses the Kullback-Leibler (KL) divergence measure between two distributions as the criterion. This criterion is optimized via a gradient-based algorithm. Other recently proposed approaches include the path alignment and part optimization to LDR [33], a manifold-learning-based technique for local linear discriminant analysis [12], and a semi-supervised approach for LDR that seeks for the best subspace on a graph-theoretic framework [13].

An approach that deserves particular attention is the one that optimizes the Bayes classification error in the transformed subspace [8, 9]. This approach, here, referred to as the Optimal Bayes LDA (OBLDA), determines the optimal linear transformation that reduces the feature space to a single dimension. It achieves this by assuming normally distributed classes with a common covariance matrix. Although an optimal solution onto a one-dimensional space is found by means of convex optimization, the approach does not readily lend itself to projections onto higher dimensional subspaces, except *via* a greedy recursive approximation algorithm that finds a solution for a d-dimensional subspace. The homoscedastic limitation is, in turn, resolved via a kernelization of the criterion. However, unless a linear kernel is used, such a strategy leads to "increasing" the dimension of the resulting subspace, rather than reducing it. The homoscedasticity limitations of FDA have also been resolved by resorting to the Approximate Pairwise Accuracy Criterion (APAC) [5]. This approach performs an *all-at-once* transformation, adds weights to the multi-class criterion, and then approximates the optimal weights using error functions on the Mahalanobis distances between the pairs of classes. All these approaches, i.e., the OBLDA, APAC, HDA, CDA and KL, have been proposed for multi-class problems, providing a solution in terms of a *single* transformation matrix that projects the data from a higher-dimensional space to a lower-dimensional space.

Finally, a few schemes have been proposed to generalize LDR methods from the two-class

to the multi-class case by perforiming and combining pairwise classifications, and they have not always been successful. Rather, they have even been reported to be less efficient than the two-class solutions. One of these schemes involves generalizing two-class classifiers to the multi-class problem by using the voting rule as proposed in [27], but this has the drawback of producing inconsequent labelings and ties. However, the authors of [27] observed these drawbacks, and attempted to solve them using the *confidence value estimation* methodology for a probabilistic voting rule so as to avoid ties. Other schemes involve one-against-all, one-against-one and all-at-once strategies, and the use of decision trees [1] for generalizing two-class Support Vector Machines (SVMs) for multi-class problems so as to avoid unclassifiable regions, inconsequent labelings, and ties. An in-depth discussion of the advantages and disadvantages of each scheme is given in [1]. Another approach that deserves special attention is the pairwise scheme proposed in [6], in which various classifiers are studied as being combined in a voting scheme.

1.5 Two-class Scenario

For two classes, we assume that their distributions have a parametric form, and whence the two classes are given in terms of their *a priori* probabilities p_1 and p_2, and two n-dimensional normally distributed random vectors, $\mathbf{x}_1 \sim N(\mathbf{m}_1; \mathbf{S}_1)$ and $\mathbf{x}_2 \sim N(\mathbf{m}_2; \mathbf{S}_2)$. The problem consists of finding a $d \times n$ transformation matrix $\mathbf{A}$ in such a way that the transformed data, given by the linear transformation $\mathbf{y} = \mathbf{A}\mathbf{x}$, becomes as separable as possible, so that it can, in turn, be classified by a relevant classification method. Various schemes have been proposed for this, and we discuss three of them here, namely the FDA, HDA and CDA.

1.5.1 The Two-class FDA Criterion

Let $\mathbf{S}_W = p_1\mathbf{S}_1 + p_2\mathbf{S}_2$ and $\mathbf{S}_E = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^t$ be the *within-class* and *between-class* scatter matrices respectively. The FDA criterion consists of finding a $d \times n$ transformation $\mathbf{A}$ that maximizes the following function [4]:

$$J_F(\mathbf{A}) = tr\left\{(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1}(\mathbf{A}\mathbf{S}_E\mathbf{A}^t)\right\} . \tag{1}$$

The matrix $\mathbf{A}$ maximizing (1), is found by solving the eigenvalue decomposition of $\mathbf{S}_F$, where

$$\mathbf{S}_F = \mathbf{S}_W^{-1}\mathbf{S}_E\,, \tag{2}$$

and by taking the d eigenvectors whose eigenvalues are the largest ones. Since $\mathbf{S}_E$ is of rank unity, $\mathbf{S}_W^{-1}\mathbf{S}_E$ is also of rank unity. Thus, the eigenvalue decomposition of $\mathbf{S}_W^{-1}\mathbf{S}_E$ leads to only a single non-zero eigenvalue, and whence FDA can only reduce to dimension $d = 1$.

1.5.2 The Two-class HDA Criterion

Loog and Duin proposed a new LDR technique for normally distributed classes [14], namely HDA, which takes into account the heteroscedasticity of the data. They considered the concept of *directed distance matrices*, and a linear transformation in the original space, to finally generalize Fisher's criterion in the transformed space by substituting the between-class scatter matrix for the corresponding directed distance matrix. The HDA criterion consists of obtaining the matrix $\mathbf{A}$ that maximizes the function [14]:

$$\begin{aligned} J_{H_{12}}(\mathbf{A}) \quad &= tr\left\{(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1} \right. \\ &\left. \left[\mathbf{A}\mathbf{S}_E\mathbf{A}^t - \mathbf{A}\mathbf{S}_W^{\frac{1}{2}}\frac{p_1\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_1\mathbf{S}_W^{-\frac{1}{2}}) + p_2\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_2\mathbf{S}_W^{-\frac{1}{2}})}{p_1p_2}\mathbf{S}_W^{\frac{1}{2}}\mathbf{A}^t\right]\right\}, \end{aligned} \tag{3}$$

where the logarithm of a matrix $\mathbf{M}$, $\log(\mathbf{M})$, is defined as:

$$\log(\mathbf{M}) = \mathbf{\Phi}\log(\mathbf{\Lambda})\mathbf{\Phi}^{-1}, \tag{4}$$

with $\mathbf{\Phi}$ and $\mathbf{\Lambda}$ representing the eigenvectors and eigenvalues of $\mathbf{M}$ respectively.

The solution to this criterion is given by the matrix $\mathbf{A}$ that is composed of the d eigenvectors (whose eigenvalues are the largest ones) of the following matrix:

$$\mathbf{S}_{H_{12}} = \mathbf{S}_W^{-1}\left[\mathbf{S}_E - \mathbf{S}_W^{\frac{1}{2}}\frac{p_1\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_1\mathbf{S}_W^{-\frac{1}{2}}) + p_2\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_2\mathbf{S}_W^{-\frac{1}{2}})}{p_1p_2}\mathbf{S}_W^{\frac{1}{2}}\right]. \tag{5}$$

1.5.3 The Two-class CDA Criterion

It has been noted in [20] that HDA considers the Chernoff distance between the classes in the original space, and incorporates this measure in the directed distance matrix to extend the FDA criterion. However, this does not guarantee that the Chernoff distance in the transformed space is maximized, and this is what is proposed in [20]. The aim of CDA is to find the matrix $\mathbf{A}$ that maximizes:

$$J^*_{C_{12}}(\mathbf{A}) = tr\left\{p_1 p_2 \mathbf{A}\mathbf{S}_E\mathbf{A}^t(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1} + \log(\mathbf{A}\mathbf{S}_W\mathbf{A}^t) - p_1\log(\mathbf{A}\mathbf{S}_1\mathbf{A}^t) - p_2\log(\mathbf{A}\mathbf{S}_2\mathbf{A}^t)\right\} \quad (6)$$

where $\mathbf{S}_W = p_1\mathbf{S}_1 + p_2\mathbf{S}_2$, and the logarithm of a matrix $\mathbf{M}$, $\log(\mathbf{M})$, is defined as in (4).

In order to maximize $J^*_{C_{12}}$, the authors of [20] proposed a gradient-based method. First, the gradient matrix is found by using the corresponding gradient operator, ∇, as follows:

$$\begin{aligned}\nabla J^*_{C_{12}}(\mathbf{A}) = \frac{\partial J^*_{C_{12}}}{\partial \mathbf{A}} = 2p_1p_2\left[\mathbf{S}_E\mathbf{A}^t(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1} - \mathbf{S}_W\mathbf{A}^t(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1}(\mathbf{A}\mathbf{S}_E\mathbf{A}^t)(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1}\right]^t \\ +2\left[\mathbf{S}_W\mathbf{A}^t(\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1} - p_1\mathbf{S}_1\mathbf{A}^t(\mathbf{A}\mathbf{S}_1\mathbf{A}^t)^{-1} - p_2\mathbf{S}_2\mathbf{A}^t(\mathbf{A}\mathbf{S}_2\mathbf{A}^t)^{-1}\right]^t. \quad (7)\end{aligned}$$

Thereafter, the algorithm finds the maximum value of the learning rate at step k, η_k, by maximizing the objective function in the direction of the gradient. The new gradient matrix at step k is obtained as $\mathbf{A}^{(k)}+\eta_k\nabla J^*_{C_{12}}(\mathbf{A}^{(k)})$, and the process is repeated until the change between the objective functions at the current and previous steps is below a user-defined threshold.

2 All-at-once Multi-class Schemes

The multi-class problem that we consider assumes c classes, $\omega_1, \ldots, \omega_c$, given by m labeled n-dimensional data points arranged in c datasets $\mathcal{D}_1 = \{\mathbf{x}_{1,1}, \ldots, \mathbf{x}_{1,m_1}\}$, ..., $\mathcal{D}_c = \{\mathbf{x}_{c,1}, \ldots, \mathbf{x}_{c,m_c}\}$. The model we consider assumes that the classes have a parametric form, and thus the c classes are given in terms of their *a priori* probabilities $p_1, \ldots, p_c$, and c n-dimensional normally distributed random vectors, $\mathbf{x}_1 \sim N(\mathbf{m}_1; \mathbf{S}_1), \ldots, \mathbf{x}_c \sim N(\mathbf{m}_c; \mathbf{S}_c)$. As these parameters are usually not known, they can be estimated from the data.

2.1 Multi-class Case

For the multi-class case, the problem consists of finding a $d \times n$ transformation matrix $\mathbf{A}$ in such a way that the transformed data, given by the linear transformation $\mathbf{y} = \mathbf{A}\mathbf{x}$, becomes as separable as possible. Three approaches that consider a *single* linear transformation matrix are the corresponding FDA, HDA and CDA methods discussed below.

2.1.1 The Multi-class FDA Criterion

The multi-class FDA criterion that we consider in this paper is the following. Suppose that $\mathbf{S}_E = \sum_{i=1}^{c} p_i(\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^t$, where $\mathbf{m} = \sum_{i=1}^{c} p_i \mathbf{m}_i$, and $\mathbf{S}_W = \sum_{i=1}^{c} p_i \mathbf{S}_i$. The aim of the FDA scheme is to find a $d \times n$ transformation matrix $\mathbf{A}$ that maximizes the criterion function given in (1), and which is obtained by finding the d eigenvectors (whose eigenvalues are the largest ones) of the matrix given in (2). Since $\mathbf{S}_E$ is of rank $r \leq c-1$, only r of these eigenvalues are nonzero, and so the FDA can, at most, only reduce to dimension $c-1$.

2.1.2 The Multi-class HDA Criterion

The HDA criterion aims to find the $d \times n$ transformation matrix $\mathbf{A}$ that maximizes [14]:

$$
\begin{aligned}
J_H(\mathbf{A}) \;=\; & \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} p_i p_j tr \Big\{ (\mathbf{A}\mathbf{S}_W\mathbf{A}^t)^{-1} \mathbf{A}\mathbf{S}_W^{\frac{1}{2}} \\
& \left[(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_{ij}\mathbf{S}_W^{-\frac{1}{2}})^{-\frac{1}{2}} \mathbf{S}_W^{-\frac{1}{2}} \mathbf{S}_{E_{ij}} \mathbf{S}_W^{-\frac{1}{2}} (\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_{ij}\mathbf{S}_W^{-\frac{1}{2}})^{-\frac{1}{2}} + \frac{1}{\pi_i \pi_j} \left(\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_{ij}\mathbf{S}_W^{-\frac{1}{2}}) \right. \right. \\
& \left. \left. -\pi_i \log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_i\mathbf{S}_W^{-\frac{1}{2}}) - \pi_j \log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_j\mathbf{S}_W^{-\frac{1}{2}}) \right) \right] \mathbf{S}_W^{\frac{1}{2}} \mathbf{A}^t \Big\} \,, \qquad (8)
\end{aligned}
$$

where $\mathbf{S}_{E_{ij}} = (\mathbf{m}_i - \mathbf{m}_j)(\mathbf{m}_i - \mathbf{m}_j)^t$, $\pi_i = \frac{p_i}{p_i + p_j}$, $\pi_j = \frac{p_j}{p_i + p_j}$, and $\mathbf{S}_{ij} = \pi_i \mathbf{S}_i + \pi_j \mathbf{S}_j$. The multi-class HDA criterion is maximized by finding the matrix $\mathbf{A}$ composed of the d eigenvectors (whose eigenvalues are the largest ones) of the following matrix:

$$
\begin{aligned}
\mathbf{S}_H \;=\; & \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} p_i p_j \mathbf{S}_W^{-1} \mathbf{S}_W^{\frac{1}{2}} \left[(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_{ij}\mathbf{S}_W^{-\frac{1}{2}})^{-\frac{1}{2}} \mathbf{S}_W^{-\frac{1}{2}} \mathbf{S}_{E_{ij}} \mathbf{S}_W^{-\frac{1}{2}} (\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_{ij}\mathbf{S}_W^{-\frac{1}{2}})^{-\frac{1}{2}} \right. \\
& \left. + \frac{1}{\pi_i \pi_j} \left(\log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_{ij}\mathbf{S}_W^{-\frac{1}{2}}) - \pi_i \log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_i\mathbf{S}_W^{-\frac{1}{2}}) - \pi_j \log(\mathbf{S}_W^{-\frac{1}{2}}\mathbf{S}_j\mathbf{S}_W^{-\frac{1}{2}}) \right) \right] \mathbf{S}_W^{\frac{1}{2}}. \qquad (9)
\end{aligned}
$$

2.1.3 The Multi-class CDA Criterion

The multi-class criterion for CDA is also an extension of the two-class case, and is obtained by maximizing the *weighted* sum of the pairwise Chernoff distances between classes ω_i and ω_j, for all $i = 1, \ldots, c-1$, $j = i, \ldots, c$, $i < j$. The weights used for the pairwise class criterion are given by the normalized joint prior probabilities between classes ω_i and ω_j, namely, $\pi_i \pi_j$. The criterion consists of finding the optimal $d \times n$ transformation $\mathbf{A}$, in such a way that the following function is maximized:

$$J_C^*(\mathbf{A}) = \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} J_{C_{ij}}^*(\mathbf{A}) \,, \tag{10}$$

where:

$$J_{C_{ij}}^*(\mathbf{A}) = tr\left\{\pi_i \pi_j (\mathbf{A}\mathbf{S}_{W_{ij}}\mathbf{A}^t)^{-1}\mathbf{A}\mathbf{S}_{E_{ij}}\mathbf{A}^t + \log(\mathbf{A}\mathbf{S}_{W_{ij}}\mathbf{A}^t) - \pi_i \log(\mathbf{A}\mathbf{S}_i\mathbf{A}^t) - \pi_j \log(\mathbf{A}\mathbf{S}_j\mathbf{A}^t)\right\}.$$

The gradient matrix, given by the first-order necessary condition, is the following:

$$\nabla J_C^*(\mathbf{A}) = \frac{\partial}{\partial \mathbf{A}} \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} J_{C_{ij}}^*(\mathbf{A}) = \sum_{i=1}^{c-1} \sum_{j=i+1}^{c} \nabla J_{C_{ij}}^*(\mathbf{A}) \,, \tag{11}$$

where:

$$\begin{aligned} \nabla J_{C_{ij}}^*(\mathbf{A}) = 2\pi_i \pi_j \left[\mathbf{S}_{E_{ij}}\mathbf{A}^t(\mathbf{A}\mathbf{S}_{W_{ij}}\mathbf{A}^t)^{-1} - \mathbf{S}_{W_{ij}}\mathbf{A}^t(\mathbf{A}\mathbf{S}_{W_{ij}}\mathbf{A}^t)^{-1}(\mathbf{A}\mathbf{S}_{E_{ij}}\mathbf{A}^t)(\mathbf{A}\mathbf{S}_{W_{ij}}\mathbf{A}^t)^{-1}\right]^t \\ +2\left[\mathbf{S}_{W_{ij}}\mathbf{A}^t(\mathbf{A}\mathbf{S}_{W_{ij}}\mathbf{A}^t)^{-1} - \pi_i\mathbf{S}_i\mathbf{A}^t(\mathbf{A}\mathbf{S}_i\mathbf{A}^t)^{-1} - \pi_j\mathbf{S}_j\mathbf{A}^t(\mathbf{A}\mathbf{S}_j\mathbf{A}^t)^{-1}\right]^t. \end{aligned} \tag{12}$$

As for the two-class case, to find the matrix $\mathbf{A}$ that maximizes $J_C^*(\mathbf{A})$, a gradient-based algorithm was proposed in [20].

3 Pairwise Multi-class Schemes

Classifiers are often developed to distinguish between just two classes of objects. A discriminant function f_{ij} is optimized such that for values larger than a certain threshold, the object is classified

as belonging to class ω_i, or otherwise belonging to class ω_j. This procedure is a direct generalization of the Bayes classifier where we estimate the density function for each class, and the object is assigned to the class with the highest posterior probability.

The principle behind designing LDR classifiers is the same. We use a linear reduction to improve the efficiency of classification by mapping the objects onto a subspace so as to apply a classifier in *that* space, which, ideally, is more suitable for classification than the original space. In this section, we use the two-class above-described well-known LDR techniques, namely the FDA, HDA and CDA, coupled with a back-end classifier, to develop the new multi-class classifiers. A generic scheme of this type is referred to as a LDR classifier (or LDRC).

When we have more than two classes, the search for the best transformed space that uses LDR methods is far more complex because of three main problems: The increase in the inter-class overlap, the decrease in the between-class separability (see Figure 1), and the existence of class covariances which are unequal. The first two handicaps can be observed in Figure 1. In (a), the classes are linearly separable in the original space and substantially overlap in the transformed space. In (b), although the classes remain to be linearly separable in the transformed space, they become closer to each other than in the original space.

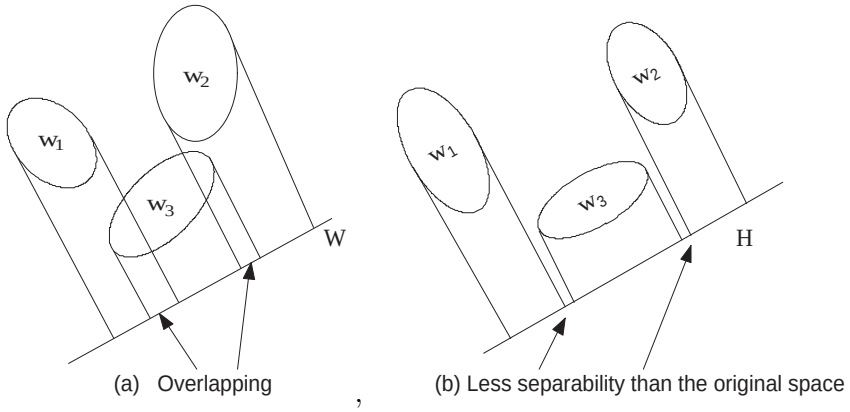

Figure 1: The effect on the overlapping between two classes and their separability by mapping them onto a lower-dimensional subspace.

For the above reasons, we need a distinct new way for treating the multi-class linear reduction problem, and in this vein, we propose three ways by which we can use two-class LDRC methods for the multi-class case, namely those that include the *Voting*, *Weighting*, and *Decision Tree* strategies respectively, all of which effectively use the set of possible two-class classifiers as shown in Figure 2. Observe that by determining the three distinct classifiers, all the pairs of classes are separable (and

even *linearly* separable) in the transformed space. The way we tackle the multi-class classification problem agrees with the scheme proposed in [6], in which it is pointed out that treating the problem as separate two-class problems has the advantage of deriving simpler classification functions, simpler decision boundaries and leading to smaller classification errors between the underlying pairs. While the starting point of our scheme is similar to the work of [6], we enhance the pairwise multi-class classification by using weighted voting and decision trees, which have lower complexity in the classification phase, and show similar classification performance, as shown later in the experimental results. Another advantage of our approach is that we consider the pairwise classification as a linear dimensionality reduction problem, while taking the heteroscedasticity of the data into account.

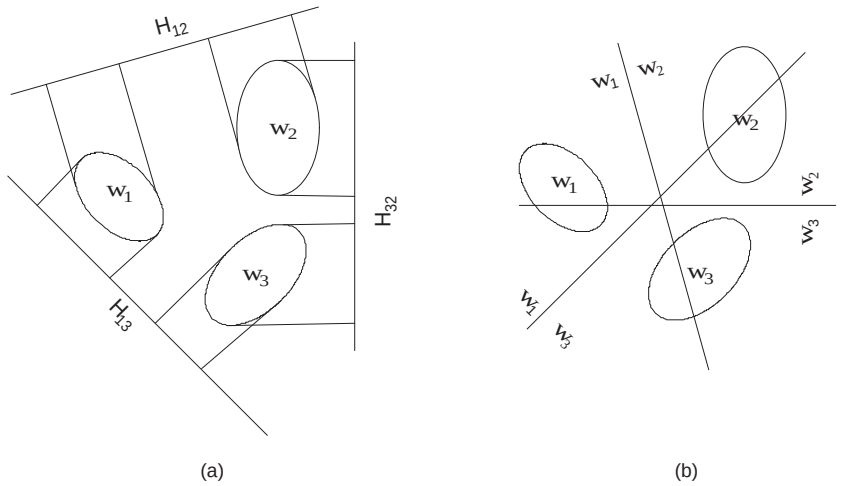

Figure 2: The various one-against-one LDR classifiers for three classes. The figure on the left side (a) is the mapping in the transformed space, while the figure on the right (b) displays the corresponding linear classifiers after using a threshold as a classifier in the transformed space.

3.1 Simple Voting

The *Simple Voting* scheme is, indeed, quite straightforward. It consists of a training and testing phase. We first train all possible two-class classifiers using the available training data, and thus obtain $\binom{c}{2}$ possible LDR classifiers. On encountering an unknown sample, $\mathbf{x}$, it is tested against all the $\binom{c}{2}$ classifiers, and every class is given a vote of unity whenever it "wins" a two-class competition. Ultimately, $\mathbf{x}$ will be labeled to the class with the highest number of votes[1].

[1]In the case of ties, the assignment is to the class with the higher *a priori* probability. If the tie still persists, a random decision is made.

3.2 Weighted Voting

The problem with the Simple Voting scheme is that ties result as a consequence of inconsistent regions (see figure 2(b)). One way of resolving this is by resorting to a *Weighted Voting* methodology, where the respective weights use the two-class posterior probabilities obtained by the LDRC.

To be more specific, as in the above case, we first train all the possible two-class classifiers using the available training data. On encountering an unknown sample, $\mathbf{x}$, it is tested against all the $\binom{c}{2}$ classifiers. For any two-class competition in which the classifier involves classes ω_i and ω_j (represented in a subspace by hyperplane H_{ij}), the confidence of $\mathbf{x}$ belonging to these classes, say $V_i(\mathbf{x})$ and $V_j(\mathbf{x})$ respectively, is increased by $f_i(\mathbf{x})$ and $f_j(\mathbf{x})$ respectively, where[2]:

$$f_i(\mathbf{x}) = \frac{P_{H_{ij}}(\omega_i|\mathbf{x})}{P_{H_{ij}}(\omega_i|\mathbf{x}) + P_{H_{ij}}(\omega_j|\mathbf{x})}, \quad \text{and} \quad f_j(\mathbf{x}) = \frac{P_{H_{ij}}(\omega_j|\mathbf{x})}{P_{H_{ij}}(\omega_i|\mathbf{x}) + P_{H_{ij}}(\omega_j|\mathbf{x})}. \tag{13}$$

In the formula given above, $P_{H_{ij}}(\omega_i|\mathbf{x})$ and $P_{H_{ij}}(\omega_j|\mathbf{x})$ are the two-class posterior probabilities of assigning $\mathbf{x}$ to ω_i and ω_j respectively as per the LDRC, and not the Bayes rule. Observe that these can be readily computed by first obtaining the means and variances in the projected space, or even using any other classification strategy, such as the nearest neighbor rule.

3.3 Decision Tree Based Fusion

From the methods described above and the literature, we see that to further improve classification for the case when ties are obtained and for unclassifiable regions, it would be advantageous to search for other ways to generalize two-class LDR classifiers for the multi-class case. To achieve this, we shall use the well-known concept of decision trees. The authors of [1] had earlier proposed the use of decision trees for the generalization of two class SVMs for the multi-class problem. In an analogous way, we show how we can utilize the same principles for LDR classifiers.

The decision trees proposed by Platt, Cristianini and Shawe-Taylor [1] generalize two-class classifiers to the multi-class case, so as to resolve classification in the unclassifiable regions. In the interest of completeness, we briefly describe the construction and application of these trees. The nodes of the tree are "Decision Boxes" which are obtained by invoking a two-class LDRC. At every node, we classify $\mathbf{x}$ based on the local discriminant function, D_{ij}, which determines whether $\mathbf{x}$ should be assigned to class ω_i or class ω_j. Without loss of generality, we assume that if $D_{ij}(\mathbf{x}) > 0$,

[2]Note that $f_i(\mathbf{x}) + f_j(\mathbf{x}) = 1$, implying that the *Simple Voting* scheme is a special 0/1 instantiation of this scenario.

$\mathbf{x}$ is assigned to class ω_i, and it is assigned to ω_j otherwise. By intelligently invoking a sequence of classifiers, one can eliminate the classes to which $\mathbf{x}$ will not be assigned, and ultimately reach a leaf node where the final classification of $\mathbf{x}$ can be achieved.

The generalization of this method to c classes is executed by list processing. The complete process to use decision trees consists of the following six steps:

1. Generate a list $\mathcal{L}$ with the indices of the classes as elements.
2. Consider the LDRC for the classes represented by the first and last elements of $\mathcal{L}$, say (i, j).
3. Calculate the value of the LDRC, $D_{ij}(\mathbf{x})$, for sample $\mathbf{x}$.
4. If $D_{ij}(\mathbf{x}) > 0$, *delete* the element j from $\mathcal{L}$. Otherwise, *delete* the element i.
5. Repeat Steps 2 to 4 until $\mathcal{L}$ has only a single element.
6. Assign $\mathbf{x}$ to the class represented by the only element in $\mathcal{L}$.

Figure 3 shows an example of such a decision tree for a 3-class problem, and the corresponding "regionalizations". One main advantage of using this scheme is that $O(c)$ decisions are made in order to classify a single sample, as opposed to the $\frac{c(c-1)}{2} = O(c^2)$ decisions needed in the voting and weighted schemes. One must note, however, that even though the ambiguous region problem is resolved, the order of the decision making nodes in the tree affect the classification performance in most of the cases. That is, applying $D_{12}(\mathbf{x})$ prior to $D_{23}(\mathbf{x})$, could discard a sample that could belong to ω_1, and is then assigned to class ω_2 or ω_3. This is because $D_{13}(\mathbf{x})$ will not be "revisited", since it is in a different branch of the tree. Thus, the *ordering* of the decision functions in the tree is an interesting problem that we are currently investigating.

4 Experimental Results

In order to evaluate the performance of the new LDRC multi-class schemes, we present an empirical analysis based on measuring the accuracies of the classifiers tested. As a benchmark for comparison, the classifiers presented here have been compared with the results obtained by the methods of [20].

The datasets that are used here are the same for all the experiments, and have been taken from the UCI ML Repository [2]. They consist of the following six datasets: Iris Plants, Pen-Based Recognition of Handwritten Digits, Thyroid Disease, Wine, Glass Identification, Vowel Recognition.

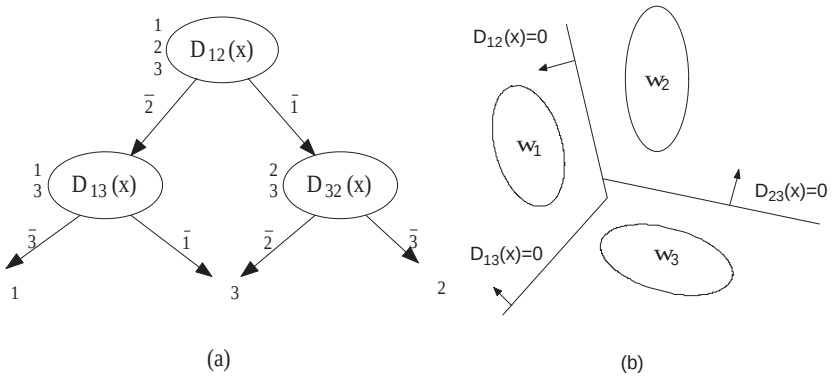

Figure 3: Decision-tree-based classification. The figure on the left (a) shows the decision-tree-based pairwise classification for three classes, while the figure on the right (b) displays the generalization regions obtained by this sequence of pairwise classifications.

Using these six datasets we performed a 10-fold cross-validation procedure. Some preprocessing was attempted to render the data applicable for our setting. Indeed, in order to avoid ill-conditioned covariance matrices, we had to apply a PCA preprocessing to the Glass Identification dataset, thus, reducing the number of dimensions from nine to eight. We also removed class '6' because it contained less than 10 elements, rendering it unsuitable for such a 10-fold cross-validation.

The comparison was made on the basis of the three LDR techniques explained above, namely the FDA, HDA and CDA methods. Also, each of these methods was complemented with linear and quadratic classifiers in the transformed space. Thus, in the tables below, we show the average of accuracy rates for the 10 folds on each dataset for the three LDR techniques coupled with their corresponding linear ($^+$L) and quadratic ($^+$Q) classifiers. In terms of nomenclature, the symbol d indicates the dimension which yielded the highest rate, n indicates the dimension of the original data, and c represents the number of classes.

For each classifier (linear and quadratic), the LDR method which achieved the highest average accuracy is marked with a '*'. A bird's eye view of these results is presented in Table 1 (a), and the comparative details are explained subsequently.

4.1 Simple Voting

In Table 1 (b), we show the results for the three LDR methods coupled with the linear and quadratic classifiers, using a *Simple Voting* strategy. There are considerable differences between the results for our benchmark, (a), and those of Simple Voting, (b), except for the Iris dataset (which differs only in FDA$^+$Q, but with a difference which is less than 1%). In the Pendigits dataset, we observe

Dataset	n	c	FDA$^+$L	d	HDA$^+$L	d	CDA$^+$L	d	FDA$^+$Q	d	HDA$^+$Q	d	CDA$^+$Q	d
(a) All-at-once:														
Iris	4	3	0.9800*	1	0.9800*	1	0.9800*	1	0.9733	1	0.9800*	1	0.9800*	1
Pendigits	16	10	0.8760*	9	0.8709	15	0.8699	15	0.9507	9	0.9768	15	0.9777*	14
Thyroid	5	3	0.9065*	1	0.9065*	4	0.9065*	1	0.9671*	1	0.9578	1	0.9626	4
Wine	13	3	0.9778	2	0.9889*	5	0.9836	2	0.9889	2	0.9945*	2	0.9945*	2
Glass	8	6	0.6613*	4	0.6032	6	0.5894	6	0.5667*	2	0.5532	4	0.5504	4
Vowel	10	11	0.5344	6	0.5556*	2	0.5556*	2	0.6212	9	0.6778	6	0.6960*	6
(b) Simple Voting:														
Iris	4	3	0.9800*	1	0.9800*	4	0.9800*	1	0.9667	1	0.9800*	2	0.9800*	3
Pendigits	16	10	0.9652	1	0.9655	16	0.9656*	2	0.9675	1	0.9813	15	0.9821*	15
Thyroid	5	3	0.9160	1	0.9483*	4	0.9483*	4	0.9626*	1	0.9578	4	0.9578	5
Wine	13	3	0.9830	1	0.9886	12	0.9889*	12	0.9889	1	0.9889	12	0.9944*	7
Glass	8	6	0.6334	1	0.6354	5	0.6586*	6	0.6080*	1	0.6051	8	0.6000	7
Vowel	10	11	0.5980	1	0.6111*	4	0.6051	9	0.6030	1	0.6990	4	0.7131*	4
(c) Weighted Voting:														
Iris	4	3	0.9800*	1	0.9800*	4	0.9800*	1	0.9667	1	0.9800*	2	0.9800*	3
Pendigits	16	10	0.9681*	1	0.9680	16	0.9681*	1	0.9692	1	0.9813	15	0.9821*	15
Thyroid	5	3	0.9160	1	0.9435*	4	0.9435*	4	0.9626*	1	0.9578	4	0.9532	1
Wine	13	3	0.9830	1	0.9886	12	0.9889*	12	0.9889	1	0.9889	12	0.9944*	7
Glass	8	6	0.6171	1	0.6213	5	0.6249*	6	0.6032*	1	0.5809	8	0.5864	7
Vowel	10	11	0.6010	1	0.6182*	5	0.6081	1	0.6040	1	0.7010	4	0.7172*	4
(d) Decision Tree:														
Iris	4	3	0.9800*	1	0.9800*	4	0.9800*	1	0.9667	1	0.9800*	2	0.9800*	3
Pendigits	16	10	0.9624	1	0.9625	16	0.9626*	2	0.9658	1	0.9813	15	0.9821*	15
Thyroid	5	3	0.9160	1	0.9483*	4	0.9483*	4	0.9626*	1	0.9578	4	0.9578	5
Wine	13	3	0.9830	1	0.9886	12	0.9889*	12	0.9889	1	0.9889	12	0.9944*	7
Glass	8	6	0.6435	1	0.6394	6	0.6589*	6	0.6080*	1	0.6051	8	0.5963	8
Vowel	10	11	0.5970	1	0.6121*	4	0.6010	1	0.6040	1	0.7030	4	0.7202*	4

Table 1: Accuracies obtained with the three all-at-once LDR methods coupled with linear and quadratic classifiers, applied on six real-life datasets from the UCI ML repository.

an improvement in all cases, especially for the linear classifier, where the average rates were 10% superior when compared to those of the All-at-once scheme. As opposed to this, in the case of the quadratic classifier, the differences are not as extensive, although the improvement was higher than 1%, which is good, considering the values are near 100%. In the Thyroid dataset we observe that the improvement in accuracy rates of the linear classifier for the three LDR methods was higher than 4% in the best case (HDA$^+$L$^+$S and CDA$^+$L$^+$S), although the quadratic classifier did not yield an enhancement in any criteria (the difference being less than 1%). It is interesting to note that in the case of the Wine dataset, we have the same values in both tables, where we attained the

same maximum values, although these maxima were reached by different approaches. Thus, for example, in the All-at-once scheme, for the linear classifier, the highest value is reached for the HDA +L (98,89%) scheme, and in Simple Voting the same maximum is attained for CDA+L+S scheme. As opposed to this, for the quadratic classifier, the highest value was obtained for the HDA+Q and CDA+Q (99.45%) methods, for which the corresponding value in Simple Voting is for CDA +Q+S (99.44%). In general, based on the results in the table, we can state that Chernoff-based classification is the most superior.

4.2 Weighted Voting

In Table 1 (c), we show the results for the three LDR methods coupled with the linear and quadratic classifiers enhanced with a *Weighted Voting* phase. Again, there are considerable differences between the results for the benchmark (a), and Weighted Voting (c), *except* for the Iris dataset, which differs only in FDA+Q in Simple Voting, although the difference is less than 1%. Also, in general, the results with *Weighted Voting* are very similar to the results of Simple Voting. In the Pendigits dataset, we observe an improvement in all cases, especially for the linear classifier average rates, which were superior by ca. 10% compared to the results of All-at-once. As opposed to this, in the case of the quadratic classifier, the differences are not as impressive, even though an improvement of more than 1% was obtained. For example, in the case of the Vowel context dataset, we have improvements in both classifiers, linear and quadratic; in the case of the linear classifier the improvements were, on the average, in all criteria by about 6%; in the case of the quadratic classifier the improvements were higher than 2% in HDA+Q+W and CDA+Q+W in Weighted Voting when compared to the HDA+Q and CDA+Q entries in Table 1 (a). Again, similar observations about the superiority of the new Chernoff-based schemes can be observed from these tables, and are not specifically re-iterated here.

4.3 Decision Tree Based Scheme

Table 1 (d) shows the results for the three LDR methods coupled with linear and quadratic classifiers invoked in conjunction with *Decision Trees*. As in the previous cases, there are considerable differences between the results for the benchmark, All-at-once, and those of the Decision Tree, except for the Iris dataset, which again differs only in the FDA+Q in Table 1 (a) with a difference which is less than 1%. In general, the results for the decision tree based methods are very similar

to those of simple and weighted voting. For example, in the Pendigits dataset, we observe an improvement in all cases, especially for the linear classifier average rates, which is ca. 10% more than what is reported in Table 1 (a). In the case of the quadratic classifier, the differences are again not as large, although the improvement is higher than 1%. An interesting behavior to observe is that the Decision Tree scheme performs as good as Simple Voting and Weighted Voting, while the time complexity that the former takes to classify an object is *linear*, against the *quadratic* complexity of the latter two schemes, where the complexity is measured on the number of classes, c.

Again, in the case of the Vowel dataset we have improvements in both classifiers, linear and quadratic; for the linear classifier, the improvements were on the average, for all criteria, by 6%, and in the case of the quadratic classifier, the improvements were less marked. In general, we infer that we can again unequivocally affirm that the Chernoff-based strategies are the most superior ones.

4.4 Comparison of All Schemes

In order to analyze the results from a different perspective, and to summarize the best results for each multi-class scheme and dataset, we provide two summarized tables of the best results. The first table shows the best results for each dataset considering all six possible LDRC schemes and the different multi-class schemes. The second table summarizes the best result for each multi-class scheme for each dataset. These results are separately discussed in the next two subsections.

4.4.1 Comparison by Dataset

Table 2 shows the best results for the three LDR methods coupled with linear and quadratic classifiers, and for the three multi-class schemes. As in the previous cases, there are the considerable differences between the results from Table 1 (a), the benchmark, and Table 2, except for the Iris dataset, which again differs only in the FDA+Q, where the difference is less than 1%. In general, the results for the best overall accuracy for all schemes from Table 2 are quite similar to the results of Table 1 (b), (c) and (d). For example, for Pendigits, we observe an improvement in all cases, especially for the linear classifier, which is ca. 10% more than what is reported in Table 1 (b). In the case of the quadratic classifier, the differences are again not as large, although the improvement is higher than 1%. As before, in the case of the Vowel dataset we observe improvements in both classifiers, the linear and quadratic; for the linear classifier, the improvements were on the average,

for all criteria, by 6%, and in the case of the quadratic classifier, the improvements were much lower. From this summarized comparison, we again confirm our earlier affirmation that the Chernoff-based LDR schemes are the most suitable for most of the datasets and for both classifiers, linear and quadratic.

Datasets	n	FDA+L	HDA+L	CDA+L	FDA+Q	HDA+Q	CDA+Q
Iris	3	0.9800*	0.9800*	0.9800*	0.9667	0.9800*	0.9800*
Pendigits	16	0.9681*	0.9680	0.9681*	0.9692	0.9813	0.9821*
Thyroid	5	0.9160	0.9483*	0.9483*	0.9626*	0.9578	0.9578
Wine	13	0.9830	0.9886	0.9889*	0.9889	0.9889	0.9944*
Glass	8	0.6435	0.6394	0.6589*	0.6080	0.6051	0.6000
Vowel	10	0.6010	0.6182*	0.6081	0.6040	0.7030	0.7202*

Table 2: Maximum average accuracies for each LDR approach coupled with the linear and quadratic classifiers, for the all-at-once method and the three one-against-one schemes.

4.4.2 Comparison by Multi-class Scheme

Table 3 shows the best results for each multi-class scheme and for each dataset, taking into account the best accuracy out of the six LDRC schemes, e.g., each LDR method combined with the linear and quadratic classifier. Here, we notice some differences between the benchmark and the one-against-one schemes, except for Iris and Wine. In Pendigits we note that the one-against-one schemes are better than the benchmark for the three schemes, where the difference is below 1%. This is quite interesting because the values are very close to 100%, and any gain should be almost insignificant. In Thyroid, the one-against-one schemes are below the benchmark, but again, the difference is very low, i.e. less than 1% – the same behavior is observed for Glass. In Vowel, the one-against-one schemes are all better than the benchmark, where the difference is more than 1%, and, for this dataset, the accuracy for the Decision Tree scheme is the highest one.

Datasets	n	Benchmark	Simple	Weighted	Tree
Iris	3	0.9800*	0.9800*	0.9800*	0.9800*
Pendigits	16	0.9777	0.9821*	0.9821*	0.9821*
Thyroid	5	0.9671*	0.9626	0.9626	0.9626
Wine	13	0.9945*	0.9944*	0.9944*	0.9944*
Glass	8	0.6613*	0.6586	0.6249	0.6589
Vowel	10	0.6960	0.7131	0.7172	0.7202*

Table 3: Maximum average accuracies for each multi-class scheme for all three LDR approaches coupled with the linear and quadratic classifiers.

Finally, if we are to submit overall concluding remarks, we can say that, in general, schemes based on Decision Trees yield the best results for datasets as the dimensionality of the feature space increases. They always gave the most superior results when there were more than 10 dimensions (except for the Iris dataset, in which we observed a tie). We also observe that all pairwise schemes improve the classification accuracy with respect to the all-at-once LDR schemes. The improvement is quite significant for the linear classifiers, while fair for the quadratic classifier. One of the reasons for this is that the quadratic classifier performs relatively well for all-at-once schemes, and whence there is not too much room for improvement. The other reason is that since the data is not necessarily normally distributed, the presence of outliers are not detected by the linear classifier which averages the covariance matrices. Additionally, another reason is that the Chernoff distance approximates very well the error rate in the transformed space, and this is what the CDA aims to do. In contrast, the FDA uses another criterion that is homoscedastic, and leads to optimal classification when the covariances are coincident – not typical in real cases. The HDA, although quite good and comparable to the CDA, does not maximize the Chernoff distance in the transformed space.

4.5 Comparison with Other All-at-once Schemes

The reader will observe that we have experimentally compared our new scheme with certain benchmark algorithms. We reckon these as the "benchmarks", because, as explained above, they all work under identical premises, rendering the "playing field to be even". We have also provided a rationale as to why we believe that these algorithms are the ones against which the comparison can be deemed to be fair.

Although an experimental comparison with other approaches has not been included here[3], in the interest of completeness, we discuss now some analytical details which will provide the reader with additional insight about the similarities and differences between our method and some of the other multi-class pairwise methods presented in this paper.

1. The OBLDA assumes that the classes are normally distributed with a common covariance matrix. The implication of this assumption is that it tacitly renders this approach to be homoscedastic. In contrast, the superior dimensionality reduction schemes introduced in this

[3] A more detailed investigation of all these methods under a host of constraints (two-class *vs.* multi-class, linear *vs.* non-linear, kernel-based *vs.* non-kernel-based, hierarchical *vs.* non-hierarchical etc.) is currently being conducted.

paper are heteroscedastic, inasmuch as HDA and CDA are also intrinsically heteroscedastic. Thus, these schemes clearly provide much more information about the classes than the OBLDA.

2. The OBLDA is able to find an *optimal solution* for the criterion only when the reduction is made onto a subspace of dimension unity. Observe that for higher dimensions, it resorts to a *greedy* recursive algorithm, whose optimality is still unproven. Thus, the OBDLA provides an "approximate" solution for minimizing the Bayes error in the transformed subspace. Note, however, that although CDA also approximates the classification error, the philosophy behind it is to optimize the *Chernoff* distance in the transformed subspace, which is a measure that has been shown to be quite accurate, even for non-normal distributions [4, 20].

3. Finally, the OBLDA is able to overcome the overlapping problem of the all-at-once schemes by means of the so-called kernel trick. To compare the schemes from this perspective, we mention that unless a *linear* kernel is used, it contradicts the fundamental principles and aims of dimensionality *reduction*, inasmuch as it usually "increases" the dimensionality when classifying, and the consequent computational burden. In all brevity, the computational complexity of classifying (that is, in the *classification phase*) in the kernelized OBLDA is polynomial in the number of dimensions, while that of Decision Trees is *linear* in both the number of classes and the number of dimensions. To clarify this, consider the setting and experiments cited in [9], which concern the Landsat dataset, and which utilize a *five*-degree polynomial kernel. Such a setting implies that the smallest dimension of the image space is $\binom{n+5}{5}$. In particular, for this specific example, since $n = 36$ and $d = 1$, it would imply a maximum of $\approx 750,000$ multiplications merely for the "reduction" phase. In contrast, our Decision Tree + CDA scheme will require $O(cn)$ classification time, and with $c = 6$, this would merely imply $6 \times 36 = 216$ multiplications. Clearly, from a computational perspective, our method performs more efficient classification than the OBLDA – we must point out that this analysis regards the *classification phase*. Using the kernel trick, however, the classification phase can be carried out by using the training samples (more specifically, the support vectors), in which case, the complexity would depend on the number of samples, the complexity of the kernel, and the dimension of the original space. In any case, the computational cost of classification would be much higher than the proposed LDR multi-class schemes.

4. To render the comparison complete, some comparative remarks against the APAC is not out of place. Within the criterion studied, the APAC can be perceived as a modification of the HDA, which rather modifies the latter by incorporating a set of *weights* for all pairs for classes. However, unlike the APAC, the HDA (which is one of the reduction methods considered here) does take into account the *heteroscedasticity* of the data when it specifically considers the information in the so-called *directed distance matrices*. Thus, in this sense, the HDA is arguably more powerful than the APAC – when viewed from the perspective of *all-at-once* schemes. Additionally, our results demonstrate that even considering the HDA in pairwise scenarios does truly lead to better classification results than the all-at-once schemes. Finally, the reader should also observe that as shown in [20], even the CDA outperforms the HDA in more cases for the set of different standard real-life datasets. This too reinforces our hypothesis that resorting to pairwise schemes is more expedient and fruitful than to all-at-once methods.

5 Conclusions and Future Work

In this paper, we have considered Linear Dimensionality Reduction (LDR) techniques for the multi-class PR problem. LDR schemes operate by invoking a relatively simple mapping of the problem onto a lower-dimensional subspace, leading to computationally efficient testing strategies. Although numerous results have been reported for the two-class problem, the corresponding issues encountered when dealing with multiple classes is far from trivial. In this paper, we have shown that it is better to solve the multi-class problem using an ensemble of Chernoff-based two-class problems, whence the overall solution is achieved by resorting to either *Voting*, *Weighting*, or to a *Decision Tree* strategy. The experimental results obtained by testing the methods on benchmark datasets demonstrate that the Chernoff-based LDR scheme works very well for one-against-one multi-class schemes. Additionally, the proposed method is not only efficient, but also yields an accuracy comparable to that obtained by the optimal Bayes classifier.

The extension of these concepts for non-linear mappings (kernel-based or otherwise) remains open. Another very interesting problem currently being investigated is that of designing algorithms to determine the *order* in which the nodes of the Decision Tree should be visited. Also, the scientific community would definitely benefit by a more detailed and comprehensive investigation of *all* these

methods under a host of constraints i.e., two-class *vs.* multi-class, linear *vs.* non-linear, kernel-based *vs.* non-kernel-based, and hierarchical *vs.* non-hierarchical. Finally, the use of other LDR criteria, such as the KL measure, in pairwise multi-class schemes is a strategy that deserves attention.

References

[1] S. Abe. *Support Vector Machines for Pattern Classification*. Springer, 2005.

[2] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. Available at http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[3] F. Camastra. Data Dimensionality Estimation Methods: A Survey. *Pattern Recognition*, 36(12):2945–2954, 2003.

[4] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. John Wiley and Sons, Inc., New York, NY, 2nd edition, 2000.

[5] R. Duin and M. Loog. Multiclass Linear Dimension Reduction by Weighted Pairwise Fisher Criteria. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(7):762–766, 2001.

[6] J. Friedman. Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University, 1996.

[7] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 1990.

[8] O. Hamsici and A. Martinez. Sparse Kernels for Bayes Optimal Discriminant Analysis. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, Minneapolis, USA, 2007.

[9] O. Hamsici and A. Martinez. Bayes Optimality in Linear Discriminant Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):647–657, 2008.

[10] A. Jain, R. Duin, and J. Mao. Statistical Pattern Recognition: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[11] S. Kim, A. Magnani, and S. Boyd. Optimal Kernel Selection in Kernel Fisher Discriminant Analysis. In *Proc. of the 23rd International Conference on Machine Learning*, pages 465–472, Pittsburgh, USA, 2006.

[12] X. Li, S. Lin, S. Yan, and D. Su. Discriminant Locally Linear Embedding with High-order Tensor Data. *IEEE Trans. on Systems, Man and Cybernetics – Part B*, 38(2):342–352, 2008.

[13] W. Liu, D. Tao, and J. Liu. Transductive Component Analysis. In *Proc. of the 8th IEEE International Conference on Data Mining*, pages 433–442. IEEE Press, 2008.

[14] M. Loog and P.W. Duin. Linear Dimensionality Reduction via a Heteroscedastic Extension of LDA: The Chernoff Criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):732–739, 2004.

[15] R. Lotlikar and R. Kothari. Adaptive Linear Dimensionality Reduction for Classification. *Pattern Recognition*, 33(2):185–194, 2000.

[16] A. Martinez and M. Zhu. Where Are Linear Feature Extraction Methods Applicable? *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(12):1934–1944, 2005.

[17] K. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Trans. on Neural Networks*, 12(2):181–202, 2001.

[18] M. Rohl and C. Weihs. Optimal vs. classical linear dimension reduction. In *Information Age, Studies in Classification, Data Analysis, and Knowledge Organization*, pages 252–259. Springer, 1999.

[19] L. Rueda. An Efficient Approach to Compute the Threshold for Multi-dimensional Linear Classifiers. *Pattern Recognition*, 37(4):811–826, April 2004.

[20] L. Rueda and M. Herrera. Linear Dimensionality Reduction by Maximizing the Chernoff Distance in the Transformed Space. *Pattern Recognition*, 41(10):3138–3152, 2008.

[21] L. Rueda and B. J. Oommen. On Optimal Pairwise Linear Classifiers for Normal Distributions: The Two-Dimensional Case. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):274–280, 2002.

[22] B. Scholkopf, S. Mika, C. Burges, P. Knirsch, K. Miiller, G. Ritsch, and A. Smola. Input Space vs Feature Space in Kernel-based Methods. *IEEE Trans. on Neural Networks*, 10(5):1000–1017, 1999.

[23] B. Scholkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. The MIT Press, 2001.

[24] B. Scholkopf, A. Smola, and K. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.

[25] D. Tao, X. Li, X. Wu, and S. Maybank. General Averaged Divergence Analysis. In *Proceedings of the 2007 Seventh IEEE International Conference on Data Mining*, pages 302–311, Omaha, USA, 2007. IEEE Press.

[26] D. Tao, X. Li, X. Wu, and S. Maybank. Geometric Mean for Subspace Selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):260–274, 2009.

[27] D. Tax and R. Duin. Using Two-Class Classifiers for Multiclass Classification. In *Proceedings of the 16th International Conference on Pattern Recognition*, volume 2, pages 124–127, Quebec, Canada, 2002.

[28] S. Theodoridis and K. Koutroumbas. *Pattern Recognition*. Elsevier Academic Press, third edition, 2006.

[29] Y. Xu, J.Y. Yang, and J. Yang. A reformative kernel fisher discriminant analysis. *Pattern Recognition*, 37(6):1299–1302, 2004.

[30] J. Yang, A. Frangi, J.Y. Yang, D. Zhang, and J. Zhong. KPCA Plus LDA: A Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(2):230–244, 2005.

[31] J. Yang and J. Y. Yang. Why Can LDA Be Performed in PCA Transformed Space? *Pattern Recognition*, 36(2):563–566, 2003.

[32] H. Yu and J. Yang. A Direct LDA Algorithm for High-dimensional Data with Application to Face Recognition. *Pattern Recognition*, 34(10):2067–2070, 2001.

[33] T. Zhang, D. Tao, X. Li, and J. Yang. Patch Alignment for Dimensionality Reduction. *IEEE Trans. on Knowledge and Data Engineering*, 2010. To appear.

[34] M. Zhu and A. Martinez. Subclass Discriminant Analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(8):1262–1273, 2006.