Simultaneous Learning of a Discriminative Projection and Prototypes for Nearest-Neighbor Classification *

Mauricio Villegas and Roberto Paredes

Universidad Politécnica de Valencia
Instituto Tecnológico de Informática
Camino de Vera s/n, 46022 Valencia (Spain)
{mvillegas,rparedes}@iti.upv.es

Abstract

Computer vision and image recognition research have a great interest in dimensionality reduction techniques. Generally these techniques are independent of the classifier being used and the learning of the classifier is carried out after the dimensionality reduction is performed, possibly discarding valuable information. In this paper we propose an iterative algorithm that simultaneously learns a linear projection base and a reduced set of prototypes optimized for the Nearest-Neighbor classifier. The algorithm is derived by minimizing a suitable estimation of the classification error probability. The proposed approach is assessed through a series of experiments showing a good behavior and a real potential for practical applications.

1. Introduction

Dimensionality reduction techniques play a very important role in image recognition tasks. Images have an inherently high dimensionality and thus it is difficult to directly apply machine learning algorithms to them because of the so called curse of dimensionality. These techniques aim at finding a mapping from the original representation space into a new space with a considerable dimensionality reduction.

Over the years several dimensionality reduction approaches have been proposed. These can be mainly categorized as: linear or non-linear, depending on the nature of the mapping function; supervised or unsupervised, depending on whether the class information is taken into account or not; parametric or non-parametric, depending on whether a distribution is assumed for the data or not; and as having a closed solution or being iterative.

Two well known linear methods are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) [10]. PCA is an unsupervised technique which preserves as much variance of the data as possible. On the other hand LDA is a supervised technique that minimizes the scatter within each class while separating them from the other classes. These two methods have a closed solution and are parametric assuming that the data has a Gaussian distribution. Subsequent versions of both techniques were presented aiming at solving some limitations of the original versions, for instance, there are non-linear extensions of PCA and LDA which rely on the kernel trick [18, 27]. Closely related to LDA is the Non-parametric Discriminant Analysis (NDA) [4, 10], which is also linear and with a closed solution, but as the name states it is non-parametric, so this method does not assume any particular distribution of the data.

Another family of methods are the techniques based on preserving the topology of the original space. Two well known methods are ISOMAP [30] and Locally Linear Embedding (LLE) [26]. Both methods are unsupervised and non-linear. A supervised version of LLE (SLLE) is presented in [8]. Also, some linear methods are proposed aiming at preserving the topology of the original space: an unsupervised method, Locality Preserving Projections (LPP) is presented in [13], and recently, a supervised method, Linear Laplacian Discrimination (LLD) is presented in [33].

The previous techniques are based on a close solution for obtaining the optimal projection. On the other hand, another family of algorithms are based on the iterative improvement of the projection under some criterion to optimize. Among others we can cite: Independent Component Analysis (ICA) [7] that can be categorized as linear and unsupervised; and Boosted Discriminant Projections [16] and Genetic Linear Projection (GLP) [23], both categorized as

*The authors would like to thank the anonymous reviewers for their careful reading and in-depth criticisms and suggestions. Work supported by the Spanish Project Consolider Ingenio 2010: MIPRCV (CSD2007-00018) and the Generalitat Valenciana - Consellería d'Educació under an FPI scholarship.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

linear and supervised.

Other dimensionality reduction methods worth mentioning are proposed in [3, 6, 9, 11, 14, 19, 28, 32].

It is important to mention that generally the dimensionality reduction techniques are independent of the classifier being used. This characteristic is good because it does not force the practitioners to use some fixed classifier, but also, it seems natural that learning both jointly, the dimensionality reduction *and* the classifier parameters, will lead to better results.

In this work we propose to use a Nearest-Neighbor (NN) classifier. The NN classifier has two parameters to estimate: the distance function and the prototypes used as reference[1]. We fix the distance function to the euclidean distance and the prototypes are estimated following the same ideas presented in the work of Paredes and Vidal [21]. Since the NN classifier operates in the projected space, the method proposed here simultaneously learns both a reduced set of prototypes and a suitable linear projection. The prototypes/projection combination is obtained by minimizing an estimation of the classification error of the NN classifier.

The proposed approach has been assessed through a series of experiments. In these experiments the algorithm exhibits a good behavior and the results shows the benefits of the simultaneous learning of the prototypes and projection base.

The rest of the paper is organized as follows: Section 2 describes the algorithm including its derivation and a discussion about its behavior. The experiments are presented in section 3 and finally section 4 draws the conclusions and directions for future research.

2. Learning Discriminative Projections and Prototypes

First, we assume that the objects of interest can be represented by elements of a D-dimensional vector space. We will refer to this space as the *original* space and denote it as $E = \mathbb{R}^D$.

Vectors in the original space can be mapped by a linear transformation to a space which we are going to refer to as the *target* space, and denote it as $G = \mathbb{R}^R$. The mapping is defined by the matrix $\mathbf{B} \in \mathbb{R}^{D \times R}$, and its columns compose the set $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_R\}$ being $\mathbf{b}_r \in \mathbb{R}^D$, $1 \leq r \leq R$. Vectors in the original space and the target space will be denoted by $\mathbf{x} \in E$ and $\mathbf{y} \in G$ respectively. The linear mapping from the original space to the target space is computed by

$$\mathbf{y} = \mathbf{B}^T \mathbf{x} \,. \tag{1}$$

[1]Generally this set of prototypes is the whole set of labeled prototypes available, training data. But it is well known that the performance of the NN rule can be boosted by using simple *editing* or *prototype reduction* techniques which attempt cleaning inter-class overlap regions

We also assume that we have a finite sample of vectors $X = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset E$ with their corresponding class labels, being C the number of classes. These samples projected onto the target space compose the set $Y = \{\mathbf{y}_1, \ldots, \mathbf{y}_N\} \subset G$. This training set is the one used for the learning of the algorithm, however, this is not the same set as the one used in the final NN classifier. For this, we define a new reduced set of prototypes $P = \{\mathbf{p}_1, \ldots, \mathbf{p}_M\} \subset E$ which has the characteristic of being much smaller than the training set, that is $M \ll N$. This set of prototypes projected onto the target space is $Q = \{\mathbf{q}_1, \ldots, \mathbf{q}_M\} \subset G$. For convenience, generally the number of prototypes per class is set to be the same for all the classes, this value will be denoted by M_c, where $M_c = M/C$. To avoid confusions throughout the rest of the paper, we clarify that when we say *training vectors* we are referring to elements of X or Y, and when we say *prototypes* we are referring to elements of P or Q.

The distance used for NN classification will be the euclidean distance between a vector and a prototype, both in the target space. This distance is given by

$$d(\mathbf{y}, \mathbf{q}) = \sqrt{\sum_{r=1}^{R} (y_r - q_r)^2} \,, \tag{2}$$

where the sub-index r denotes the rth component.

2.1. Derivation of the Algorithm

The objective is to use a training set X to obtain a discriminant projection base B and a reduced set of prototypes P that produce a low error rate of the NN classifier in the target space. For this, we propose to minimize a criterion index which is an approximation to the NN classification error of X using P and $d(.,.)$. Following a similar notation as in [21], this NN error estimate can be written as

$$J(B, P) = \frac{1}{N} \sum_{\forall \mathbf{x} \in X} S_\beta \left(\frac{d(\mathbf{y}, \mathbf{q}^=)}{d(\mathbf{y}, \mathbf{q}^\neq)} \right) \,, \tag{3}$$

where $\mathbf{y} = \mathbf{B}^T \mathbf{x}$ and $\mathbf{q}^=$, $\mathbf{q}^\neq \in Q$ are, respectively, the same-class and different-class nearest prototypes of $\mathbf{y}$. Each of these prototypes have a corresponding vector in the original space, which we denote by $\mathbf{p}^=$ and $\mathbf{p}^\neq$ respectively. Note that $\mathbf{p}^=$ and $\mathbf{p}^\neq$ are not necessarily the same-class and different-class nearest prototypes in the original space. The function $S_\beta(z)$ is a sigmoid with slope β centered at $z = 1$, which is defined as

$$S_\beta(z) = \frac{1}{1 + e^{\beta(1-z)}} \,. \tag{4}$$

Note that as β approaches infinity, the sigmoid function tends to the step function, and thus the index J is an estimation of the NN classification error using $\mathbf{B}$ and $\mathbf{P}$. However,

by using a sigmoid function, the index becomes differentiable. Furthermore the sigmoid function has a smoothing effect that is beneficial for the behavior of the algorithm. For further details on this approximation to the NN classification error refer to [22].

A gradient descent procedure is proposed to minimize this index. This requires to take partial derivatives of J with respect to the parameters being optimized B, and P (which indirectly optimizes Q). It should be noted that J depends on B and P through the distance $d(.,.)$ in two different ways. First, it depends directly through the projection base and prototypes involved in the definition of $d(.,.)$. The second, more subtle dependence is due to the fact that for some $\mathbf{y} \in Y$, the nearest prototypes $\mathbf{q}^=$ and $\mathbf{q}^\neq$ may change as the projection base and prototype positions are varied.

While the derivatives due to the first dependence can be developed from equation (3), the secondary dependence is non-continuous and is thus more problematic. Therefore a simple approximation will be followed here by assuming that the secondary dependence is not significant compared to the first one. In other words, we will assume that, for sufficiently small variations of the projection base and prototype positions, the prototype neighborhoods remain unchanged. Correspondingly, we can derive from equations (2) and (3) the following:

$$\frac{\partial J}{\partial \mathbf{b}_r} \approx \frac{1}{N} \sum_{\forall \mathbf{x} \in X} \frac{S'_\beta\left(f(\mathbf{y})\right) f(\mathbf{y})}{d^2(\mathbf{y}, \mathbf{q}^=)} (y_r - q_r^=)(\mathbf{x} - \mathbf{p}^=) \quad ; \quad (5)$$
$$- \frac{1}{N} \sum_{\forall \mathbf{x} \in X} \frac{S'_\beta\left(f(\mathbf{y})\right) f(\mathbf{y})}{d^2(\mathbf{y}, \mathbf{q}^\neq)} (y_r - q_r^\neq)(\mathbf{x} - \mathbf{p}^\neq)$$

$$\frac{\partial J}{\partial \mathbf{p}_m} \approx \frac{1}{N} \sum_{\substack{\forall \mathbf{x} \in X: \\ \mathbf{q}_m = \mathbf{q}^\neq}} \frac{S'_\beta\left(f(\mathbf{y})\right) f(\mathbf{y})}{d^2(\mathbf{y}, \mathbf{q}^\neq)} \sum_{r=1}^{R} (y_r - q_r^\neq)\mathbf{b}_r \quad . \quad (6)$$
$$- \frac{1}{N} \sum_{\substack{\forall \mathbf{x} \in X: \\ \mathbf{q}_m = \mathbf{q}^=}} \frac{S'_\beta\left(f(\mathbf{y})\right) f(\mathbf{y})}{d^2(\mathbf{y}, \mathbf{q}^=)} \sum_{r=1}^{R} (y_r - q_r^=)\mathbf{b}_r$$

As it was mentioned before, the super-indexes $^=$ and $^\neq$ indicate that the prototype is the same-class or different-class nearest prototypes of $\mathbf{y}$ respectively. The function $f(\mathbf{y})$ is the ratio of the distances to the same-class and different-class nearest prototypes, that is

$$f(\mathbf{y}) = \frac{d(\mathbf{y}, \mathbf{q}^=)}{d(\mathbf{y}, \mathbf{q}^\neq)} \, , \quad (7)$$

Algorithm LDPP $(X, B, P, \beta, \gamma, \eta, \varepsilon)$ {
// X: training set; B, P: initial parameters;
// β: sigmoid slope; γ, η: learning factors; ε: small constant;
$\lambda' = \infty$; $\lambda = J(B, P)$; $B' = B$; $P' = P$;
while$(|\lambda' - \lambda| > \varepsilon)$ {
$\lambda' = \lambda$;
for $m = 1 \ldots M$
$\mathbf{q}_m = \mathbf{B}^T \mathbf{p}_m$;
for all $\mathbf{x} \in X$ {
$\mathbf{y} = \mathbf{B}^T \mathbf{x}$;
$\mathbf{q}^= = \text{FINDNNSAMECLASS}(Q, \mathbf{y})$;
$\mathbf{q}^\neq = \text{FINDNNDIFFCLASS}(Q, \mathbf{y})$;
$F^= = S'_\beta\left(f(\mathbf{y})\right) f(\mathbf{y}) / d^2(\mathbf{y}, \mathbf{q}^=)$;
$F^\neq = S'_\beta\left(f(\mathbf{y})\right) f(\mathbf{y}) / d^2(\mathbf{y}, \mathbf{q}^\neq)$;
for $r = 1 \ldots R$ {
$\mathbf{b}'_r = \mathbf{b}_r - F^=(y_r - q_r^=)(\mathbf{x} - \mathbf{p}^=)$;
$\mathbf{b}'_r = \mathbf{b}_r + F^\neq(y_r - q_r^\neq)(\mathbf{x} - \mathbf{p}^\neq)$;
$\mathbf{p}^{=\prime} = \mathbf{p}^= + F^=(y_r - q_r^=)\mathbf{b}_r$;
$\mathbf{p}^{\neq\prime} = \mathbf{p}^\neq - F^\neq(y_r - q_r^\neq)\mathbf{b}_r$;
}
}
$B = B'$; $P = P'$; $\lambda = J(B, P)$;
}
return(B, P);
}

Figure 1. Learning discriminant projections and prototypes (LDPP) algorithm.

and S'_β is the derivative of the sigmoid function (4) with respect to z

$$S'_\beta(z) = \frac{\beta e^{\beta(1-z)}}{(1 + e^{\beta(1-z)})^2} \, . \quad (8)$$

Using the derivatives in equations (5) and (6) leads to the corresponding gradient descent update equations

$$\mathbf{b}_r^{(t+1)} = \mathbf{b}_r^{(t)} - \gamma \frac{\partial J}{\partial \mathbf{b}_r} \, , \quad (9)$$

$$\mathbf{p}_m^{(t+1)} = \mathbf{p}_m^{(t)} - \eta \frac{\partial J}{\partial \mathbf{p}_m} \, . \quad (10)$$

2.2. LDPP Algorithm

The gradient descent procedure is summarized in the algorithm Learning Discriminant Projections and Prototypes (LDPP), see figure 1. From a dimensionality reduction point of view, it can be categorized as being linear, supervised and non-parametric.

The algorithm procedure is somewhat intuitive. The proposed algorithm visits each vector $\mathbf{x} \in X$ in each iteration and updates the projection base and the prototype positions. The projection base is modified so that it projects the vector $\mathbf{x}$ close to its same-class nearest prototype in the target

space, $\mathbf{q}^{=}$. Similarly, the projection base is also modified so that it projects the vector $\mathbf{x}$ far away from its different-class nearest prototype in the target space, $\mathbf{q}^{\neq}$. Simultaneously, the prototypes of the reduced set Q are modified in the following way: for each vector $\mathbf{x} \in X$, its same-class nearest prototype in the target space $\mathbf{q}^{=}$ is moved towards the projection of $\mathbf{x}$, while its different-class nearest prototype $q^{\neq}$ is moved away from the projection of $\mathbf{x}$. This desirable movements in the target space are accomplished by the update of the prototypes in the original space.

We can discuss the problems and disadvantaged that the algorithm has. First of all, it is iterative and a gradient descent method will only guarantee finding a local minimum. The solution will depend on the initialization of the parameters and the learning rates chosen. Finally, in certain tasks the data distributions could be naturally non-linear, and therefore using a linear projection could be an important disadvantage.

On the other hand, the algorithm has several interesting properties. The index being minimized is directly related to the NN classification error, and therefore it is directly related to the final classifier to use. The update steps are weighted by the distance ratio $f(\mathbf{y})$ and windowed by the derivative of the sigmoid function. This way, only the training vectors which are close to the decision boundaries actually contribute to the update of the parameters. A suitable β value of the sigmoid function should allow the proposed algorithm to learn from the prototypes that lay near the class decision boundaries, moreover, the windowing effect of the sigmoid derivative should prevent learning from outliers whose $f(\mathbf{y})$ value is too large and also should prevent learning from those vectors that are safely well classified (with $f(\mathbf{y}) \ll 1.0$). The proposed algorithm also condenses the training set into a very compact classifier, and this added to the fact that it is linear, makes the classifier extremely fast. In this sense the learned classifier is also expected to generalize well to unseen data thanks to the condensation of the data and the effect of the derivative of the sigmoid function just mentioned.

It is important to clarify that the projection base $\mathbf{B}$ is not forced to be an orthonormal basis. Also there is no upper limit in the number of dimensions of the target space, unlike other discriminative techniques which have a $C-1$ upper bound [10], and even it is not mandatory that the dimensionality of the target space R be smaller than the original space D.

3. Experiments

The capabilities of the proposed approach have been empirically assessed through three different types of experiments. In the first one, a handwritten digits recognition task is considered. In the second one a more challenging experiment on Gender Recognition is proposed over an extensive data set. Finally, the last experiment is carried out over a Face Recognition task. This last experiment will show the generalization capabilities of the projection base obtained by LDPP.

3.1. Handwritten Digits

This experiment is basically the same as the one found in the work of He and Niyogi [13]. It was conducted using the *Multiple Features Database* [1] which is a data set of features of handwritten digits ('0'–'9') extracted from a collection of Dutch utility maps. This data set comprises 200 binary images per class (for a total of 2,000 images). Each image is represented by a 649-dimensional vector that includes: 76 Fourier coefficients of the character shapes; 216 profile correlations; 64 Karhunen-Love coefficients; 240 pixel averages in 2×3 windows; 47 Zernike moments; and 6 morphological features.

With this corpus and using the LDPP algorithm, a 2-dimensional projection base and one prototype per class was learned. The corpus was not divided into training and test sets and all of the data was used for the learning. This is because the objective of the experiment is just observing in a 2-D graph the resulting representation of the data.

The initialization of the algorithm was: the first two PCA components for the projection base; and the class mean for each prototype. The β parameter was kept fixed to 10 and the learning factors chosen were $\gamma = 10$ and $\eta = 0.1$. The algorithm was executed for 10000 iterations. The results are presented in figure 2. This figure shows two graphs that plot the prototype of each class in the target space with the corresponding voronoi diagram that they produce. To see the relationship with the prototypes learned, the training set is also plotted in the graphs. The first graph shows the initialization, and the second one is the final result obtained with the algorithm. For comparison of these results with other techniques on the same data see [13, 23].

This experiment is quite illustrative for showing the capabilities that the LDPP algorithms offers. Although the initialization is not very good, as can be observed in the figure, the algorithm is able to find a projection base that nicely groups each class making them almost separable. It is worth mentioning that the graphs in the figure have the same scale in the horizontal and vertical axis, and it is interesting to note that for LDPP the variance of the data in both axis is very similar, therefore the two components have more or less the same importance. In fact it is like if the data points within each class were whitened, which is exactly what is needed if the euclidean distance is used for the NN.

3.2. Gender Recognition

Although there are several works on gender recognition of human face images [4, 5, 12, 16], there is no standard database or protocol for experimentation in this task. For

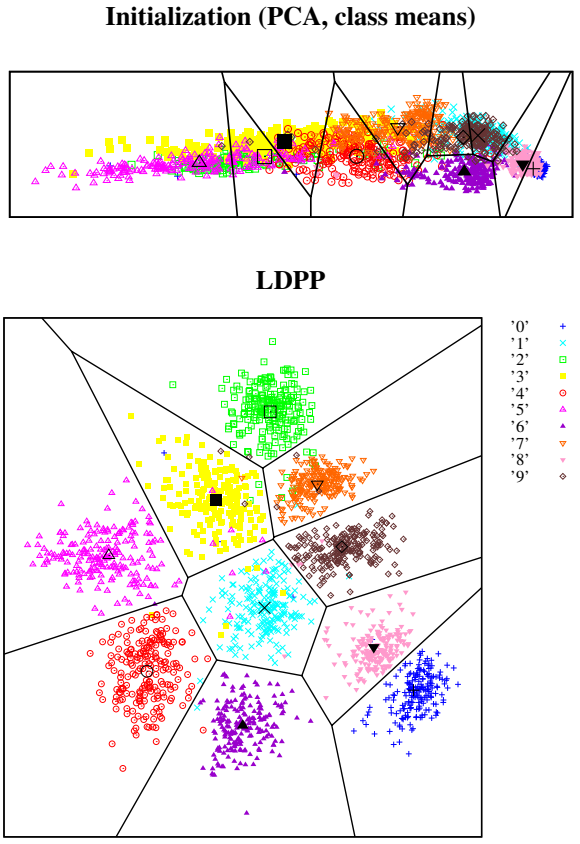

Figure 2. Prototypes plotted in the target space for handwritten digits and the corresponding voronoi diagram. Top graph is the initialization (PCA and class means) and bottom, the final result learned with LDPP. The graphs also include the training set points projected onto the target space.

our experiments we have taken a set of 1892 images (946 males and 946 females) from many databases. From each database we took only the first frontal image of each subject, however because all of the databases have more male subjects than female, we only took as many male images as there were females. More specifically the data set was composed of the following images: 118 from the AR Face Database [15]; 82 from the BANCA Database [2]; 22 from the Caltech Frontal Face Database [31]; 102 from the Essex Collection of Facial Images [29]; 792 from the FERET Database [25]; 486 from the FRGC version 2 Database [24]; 14 from the Georgia Tech Face Database [20]; and 276 from the XM2VTS Database [17].

The preprocessing done to the images was as follows. Using manually selected eye coordinates, the face images were cropped and resized to 32×40. Afterward, the images were converted to gray-scale and histogram equalized in order to somewhat compensate for global illumination changes. This gives a 1280-dimensional vector representation of each image which was what we used for the experiments.

A five-fold cross validation procedure was employed. In this procedure the data set is randomly divided into 5 subsets, four subsets are used for training and the remaining one for test. The experiments are repeated each time using a different subset for test and the results are averaged.

The LDPP algorithm was used to learn a projection base and a set of prototypes. The number of dimensions of the target space and the number of prototypes per class was varied, $R = \{1, 2, 4, 8, 16, 32\}$ and $M_c = \{1, 2, 4, 8, 16\}$. The initialization of the projection base was by using PCA and for the prototypes the class mean (if there are several prototypes per class, they are initialized to the class mean randomly perturbed). The β parameter was kept fixed to 10 and the learning factors chosen were $\gamma = 0.5$ and $\eta = 1000000$. The algorithm was executed for 10000 iterations.

Approach	**Dim.**	**Error (%)**		**Classif. Time (relative)**
		Mean	**Std. Dev.**	
Orig. Space	1280	21.1	1.8	1
PCA	64	20.2	2.5	$\approx 10^{-1}$
LDA	1	29.0	2.6	$\approx 10^{-2}$
MFA [6]	1	30.9	3.4	$\approx 10^{-2}$
CPW [22]	1280	17.4	1.1	≈ 1
LPP [13]	1	15.6	2.5	$\approx 10^{-2}$
LPD [21]	1280	13.3	1.4	$\approx 10^{-2}$
SVM	N/A	12.0	1.3	$\approx 10^{-1}$
LDPP$_{(M_c=1)}$	1	11.6	1.7	$\approx 10^{-4}$
LDPP$_{(M_c=4)}$	2	10.6	1.6	$\approx 10^{-3}$
LDPP$_{(M_c=16)}$	32	**9.5**	**1.2**	$\approx 10^{-2}$

Table 1. Gender recognition results comparing baseline techniques with LDPP.

Table 1 compares a few results for LDPP with some baseline techniques. For the baseline techniques, the parameters were also varied and in the table we only show the best results. Not including SVM, all of the results in the table use the NN classifier. The vectors used as reference for the NN classifier are the whole training set except for LDPP and LPD that use a set of learned prototypes. The result for SVM was obtained using SVMLight, for CPW and LPD using R. Paredes's [21] implementation and for MFA and LPP using D. Cai's [6] implementation.

In general, the results for LDPP for all the parameters tried were very good and with similar error rates. In the table we only show three representative results. The best error obtained was for $M_c = 16$ and $R = 32$, and it is considerably better than the baseline techniques. The methods that reduce dimensionality to only one dimension tend to give high error rates, which is probably due to such a drastic reduction. Nonetheless in this situation LDPP still gives a competitive result. The table also includes the average classification time relative to the time for the original space.

As the dimensionality and the number of reference vectors of the NN classifier are lower, the classification time also decreases.

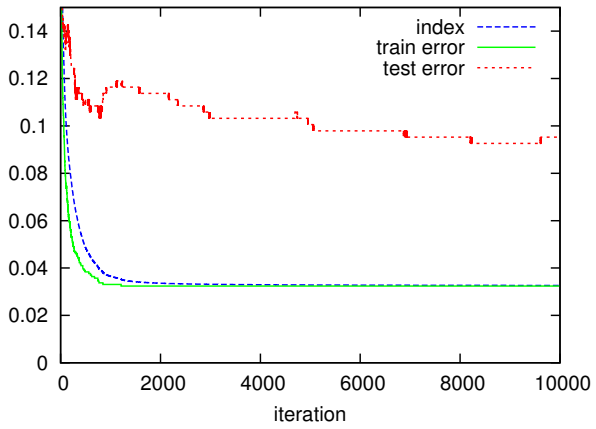

Figure 3. Graph of the index J and the training and test set errors as the LDPP algorithm iterates. This is for the gender recognition experiment with $R = 16$ and $M_c = 4$.

Figure 3 shows a graph of the index J and the classification error of the training and test sets as they vary with the iterations of the LDPP algorithm. The index J is an approximation of the training set error, and as expected, in the graph these two are always very close together. On the other hand, the test set error tends to be higher than the train set error. It is important to note that even when the training set reaches a lower bound (iteration 2000) and the algorithm is further iterated, the test set error does not increase. In fact the error rate goes on decreasing. This is a very important property, the algorithm does not suffer much by over-fitting. This is mainly due to the basic idea of the algorithm of having a low dimensionality and a small set of prototypes that tend to a good generalization capability.

Thanks to the fact that the feature vectors are images, the projection base and prototypes learned can also be viewed as images. An example of this is shown in figure 4 for a target space of 2 dimensions and 2 prototypes per class. In the figure, the prototypes and the training set are also plotted in a 2-D graph showing how the prototypes define a fair decision boundary for the classes. The two projection vectors have the appearance of strange faces which encode the discriminative features of the genders, however it is difficult to really interpret them.

3.3. Face Recognition

Probably all of the techniques for discriminative dimensionality reduction (including LDPP proposed in this paper) have as objective to find a space in which the classes of the training set are well separated. However in some applications, the real objective is somewhat different. In face

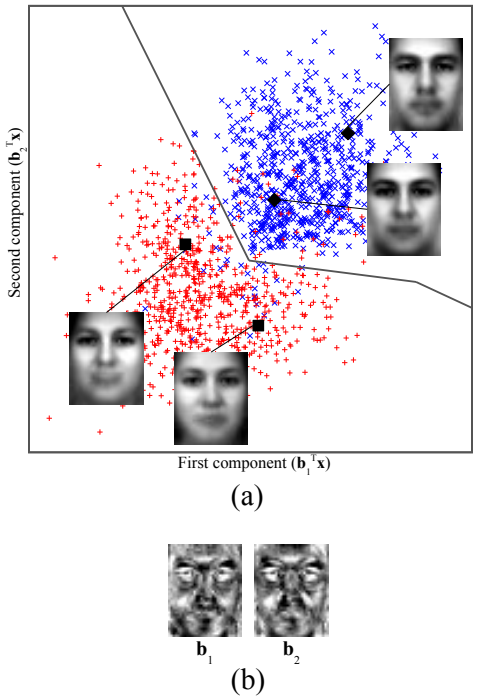

Figure 4. (a) Prototypes for gender recognition learned with LDPP for $R = 2$ and $M_c = 2$. For the original dimensionality the prototypes are represented as images, and for the target space they are plotted in a 2-D graph. The 2-D graph also includes the voronoi diagram and the training set points. (b) The two projection vectors represented as images.

recognition there can be a variable number of classes (people) and therefore it is desired to train the system with images of people different from the final users. To this end, a learned projection base should discriminate faces of people in general, not only for the subjects in the training set. The LDPP algorithm is not designed for this type of problems because a set of reference prototypes is also learned and such set is optimized for the training used. However we are going to show in this experiment that the projection base learned is adequate for being used without the prototypes.

This experiment is a face identification task, and it is the same as the one found in the work of Zhao *et al*. [33]. The images used are a subset of the facial data in experiment 4 of FRGC version 2 [24]. There are in total 316 subjects, each one having ten images. The first 200 subjects are used for the test phase, using the first five images of each subject for the gallery set and the remaining images for the probe set. All the images from the last 116 subjects are used for the training set, which is the one for leaning the projection base. Using the eye coordinates, the images were cropped and resized to 32×40. The images were also converted to gray-scale and there was no illumination normalization. The data set and the experimentation protocol are clearly

explained in [33].

The procedure of the experiment is as follows. The training set is used to learn a projection base. This projection base is used to dimensionally reduce the gallery and probe sets, and a classification error estimation is obtained using the nearest neighbors of the gallery set. In this procedure the prototypes obtained from LDPP are never used, they are discarded after the training phase.

Figure 5 shows a graph of the face identification error varying the number of dimensions of the target space. The algorithm was iterated 5000 times using one prototype per class, $\beta = 10$, $\gamma = 0.5$ and $\eta = 100$. The initialization was using PCA for the projection base and the class means for the prototypes. The graph shows a curve for the initialization (PCA) and another curve for the final result with LDPP.

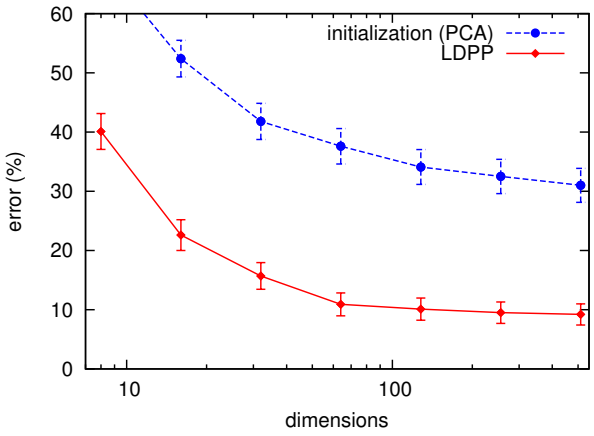

Figure 5. Face identification error varying the number of dimensions for the initialization by using PCA and the final result for LDPP (discarding the prototypes). The error bars indicate the 95% confidence intervals.

The algorithm consistently improves the projection base with respect to the initialization. The lowest error obtained was 9.2% $\pm$1.8 for 512 dimensions, which is competitive compared to other techniques [33], see table 2. Then effectively the projection base learned with LDPP is a discriminative projection adequate to be used without the prototypes obtained.

Approach	Error (%)
Laplacianfaces	15.0
L-Fisherfaces	9.5
LBP plus Dual LLD	7.4
Initialization (PCA)	31.0
LDPP	9.2

Table 2. Face recognition results comparing baseline techniques with LDPP.

4. Conclusions

In this paper we have proposed a novel algorithm which simultaneously optimizes a linear discriminant projection base, adequate for dimensionality reduction, and a reduced set of prototypes for Nearest-Neighbor classification. First the algorithm was formally derived based on a minimization of a suitable estimation of the classification error probability. Afterward, the characteristics, strengths and weaknesses of the algorithm were analyzed theoretically and through a series of practical experiments. Based on the results of the experiments, we can conclude that the proposed approach is capable of giving a good performance for a great variety of problems. Furthermore, the learned classifiers are very compact and simple, making them a good choice from a practical point of view.

The algorithm was carefully studied in the present paper, however there are still some topics that need to be further analyzed. In all of the experiments the initialization of the algorithm was done using PCA for the projection base and class means for the prototypes. This is not necessarily a good initialization for every problem, and by using other methods it is possible to obtain better results and a faster convergence. Another topic of study is concerning the learning rates. Currently these are chosen by a trial and error procedure and there is no clear relationship between the learning rates of the projection base and the prototypes. An in-depth study can give a general behavior of these parameters and provide a simple way for selecting them depending on the characteristics of the task.

Although the technique was tested in a few tasks achieving successful results, we believe that it has a high practical value and we want to test it for several other applications. To name a few there are: classifier combination, biometric fusion, face verification and face expression analysis. Finally, another direction for future research will be in the development of new algorithms based on the same ideas. The obvious following step is developing a non-linear extension of the proposed algorithm, for instance, using the kernel trick.

References

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

[2] E. Bailly-Bailliére, S. Bengio, F. Bimbot, M. Hamouz, J. Kittler, J. Mariéthoz, J. Matas, K. Messer, V. Popovici, F. Porée, B. Ruíz, and J.-P. Thiran. The BANCA database and evaluation protocol. In *AVBPA*, pages 625–638, 2003.

[3] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *NIPS*, pages 585–591, 2001.

[4] M. Bressan and J. Vitrià. Nonparametric discriminant analysis and nearest neighbor classification. *Pattern Recognition Letters*, 24(15):2743–2749, 2003.

[5] S. Buchala, N. Davey, R. Frank, and T. Gale. Dimensionality reduction of face images for gender classification. *Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference*, 1:88–93 Vol.1, 22-24 June 2004.

[6] D. Cai, X. He, Y. Hu, J. Han, and T. Huang. Learning a spatially smooth subspace for face recognition. *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–7, 17-22 June 2007.

[7] J.-F. Cardoso and P. Comon. Independent component analysis, a survey of some algebraic methods. In *Proc. ISCAS'96*, volume 2, pages 93–96, 1996.

[8] D. de Ridder, O. Kouropteva, O. Okun, M. Pietikäinen, and R. P. W. Duin. Supervised locally linear embedding. In *ICANN*, pages 333–341, 2003.

[9] D. de Ridder, M. Loog, and M. J. T. Reinders. Local fisher embedding. In *ICPR (2)*, pages 295–298, 2004.

[10] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, 2 edition, 1990.

[11] A. Globerson and S. T. Roweis. Metric learning by collapsing classes. In *NIPS*, 2005.

[12] A. B. A. Graf and F. A. Wichmann. Gender classification of human faces. In *BMCV '02: Proceedings of the Second International Workshop on Biologically Motivated Computer Vision*, pages 491–500, London, UK, 2002. Springer-Verlag.

[13] X. He and P. Niyogi. Locality preserving projections. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[14] A. Jianchang Mao; Jain. Artificial neural networks for feature extraction and multivariate data projection. *Neural Networks, IEEE Transactions on*, 6(2):296–317, Mar 1995.

[15] A. Martinez and R. Benavente. The AR face database. CVC Technical Report #24, June 1998.

[16] D. Masip and J. Vitrií. Boosted discriminant projections for nearest neighbor classification. *Pattern Recognition*, 39(2):164–170, 2006.

[17] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: The extended M2VTS database. In R. Chellapa, editor, *Second International Conference on Audio and Video-based Biometric Person Authentication*, pages 72–77, Washington, USA, Mar. 1999. University of Maryland.

[18] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

[19] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, A. Smola, and K.-R. Müller. Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(5):623–633, 2003.

[20] A. V. Nefian. Georgia tech face database. http://www.anefian.com/face_reco.htm.

[21] R. Paredes and E. Vidal. Learning prototypes and distances: a prototype reduction technique based on nearest neighbor error minimization. *Pattern Recognition*, 39(2):180–188, 2006.

[22] R. Paredes and E. Vidal. Learning weighted metrics to minimize nearest-neighbor classification error. *IEEE Transaction on Pattern Analisys and Machine Intelligence*, 28(7):–, 2006.

[23] A. J. Perez-Jimenez and J. C. Perez-Cortes. Genetic algorithms for linear feature extraction. *Pattern Recognition Letters*, 27(13):1508–1514, 2006.

[24] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek. Overview of the face recognition grand challenge. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 947–954, Washington, DC, USA, 2005. IEEE Computer Society.

[25] P. J. Phillips, H. Moon, S. A. Rizvi, and P. J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[26] S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, Dec. 2000.

[27] B. Schölkopf, A. J. Smola, and K.-R. Müller. Kernel principal component analysis. *Advances in kernel methods: support vector learning*, pages 327–352, 1999.

[28] A. Sharma, K. Paliwal, and G. Onwubolu. Class-dependent pca, mdc and lda: A combined classifier for pattern classification. 39(7):1215–1229, July 2006.

[29] L. Spacek. Essex collection of facial images. http://cswww.essex.ac.uk/mv/allfaces/index.html.

[30] J. B. Tenenbaum, V. de Silva, , and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[31] M. Weber. Caltech frontal face database. http://www.vision.caltech.edu/html-files/archive.html.

[32] T. Zhang, Sheng; Sim. Discriminant subspace analysis: A fukunaga-koontz approach. *Transactions on Pattern Analysis and Machine Intelligence*, 29(10):1732–1745, Oct. 2007.

[33] D. Zhao, Z. Lin, R. Xiao, and X. Tang. Linear laplacian discrimination for feature extraction. In *CVPR07*, pages 1–7, 2007.