Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition 39 (2006) 164–170
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Boosted discriminant projections for nearest neighbor classification

David Masip, Jordi Vitrià*

Computer Vision Center, Dept. Informàtica, Universitat Autònoma de Barcelona, Bellaterra, Spain

Abstract

In this paper we introduce a new embedding technique to find the linear projection that best projects labeled data samples into a new space where the performance of a Nearest Neighbor classifier is maximized. We consider a large set of one-dimensional projections and combine them into a projection matrix, which is not restricted to be orthogonal. The embedding is defined as a classifier selection task that makes use of the AdaBoost algorithm to find an optimal set of discriminant projections. The main advantage of the algorithm is that the final projection matrix does not make any global assumption on the data distribution, and the projection matrix is created by minimizing the classification error in the training data set. Also the resulting features can be ranked according to a set of coefficients computed during the algorithm. The performance of our embedding is tested in two different pattern recognition tasks, a gender recognition problem and the classification of manuscript digits.

Keywords: Feature extraction; Classifier selection; Linear discriminant analysis; Boosting; Prototype selection; Dimensionality reduction

1. Introduction

This paper deals with feature extraction applied to nearest neighbor classification. Feature extraction allows a compact representation of the input data, due to the dimensionality reduction achieved during the process, what increases the performance of the global scheme reducing the storage needs and the computational costs. In our case we have focused on discriminant analysis techniques, which take into account class membership of the input data, learning invariant characteristics that increase the classification ratios.

Maybe one of the first attempt to dimensionality reduction applied to classification is principal component analysis [1,2] where the goal is to find the linear projection matrix that preserves the maximum amount of input data variance. In discriminant analysis the labels are also considered in the linear feature extraction process, and the goal is to find the orthogonal set of basis that maximizes some separability criteria. The main problem of linear discriminant algorithms is their dependency on a set of assumptions that sometimes are not met in the data distribution [3].

Last years some nonlinear algorithms applied to feature extraction have appeared. Tenenbaum et al. [4] introduced the isomap algorithm, which tries to preserve the geodesic distances between points in the low-dimensional embedding. Roweis et al. [5] introduced a new nonlinear technique that preserves the local neighborhood of each point in the embedding process. The nonlinear nature of both techniques allows to represent the manifold that underlay the training samples, but there are some difficulties using both algorithms with new unseen input vectors. Also the features extracted using this nonlinear techniques cannot be ranked in order of importance for classification purposes.

What we purpose here is an embedding from high-dimensional space to a low-dimensional one, where the features are ranked according to coefficients computed within the algorithm. Also we have not made assumptions

* Corresponding author. Tel.: +34 93 581 1828.
E-mail addresses: davidm@cvc.uab.es (D. Masip), jordi@cvc.uab.es (J. Vitrià).

0031-3203/$30.00
doi:10.1016/j.patcog.2005.06.004

on the data distributions, and we do not force our projection to be orthogonal [6]. Our embedding combines a set of simple 1D projections, which can complement each other to achieve better classification results. We have made use of AdaBoost algorithm as a natural way to select the feature extractors, and the coefficients that can rank the importance of each projection.

2. Feature extraction for classification

The main goal of this work is to find a mapping from a high-dimensional space to new one that optimizes a discriminability criteria on the input data that is suited for nearest neighbor classification. Discriminant analysis can be very useful for this task. In this section we will review the classic Fisher discriminant analysis (FLD), and an evolution of the algorithm introduced by Fukunaga and Mantock [7], the non parametric discriminant analysis (NDA), which improves the classification results by using the nearest neighbor classifier and also overcomes the two main drawbacks of FLD:

- Gaussian assumption over the class distribution of the data samples.
- Dimensionality of the subspaces obtained which is limited by the number of classes.

2.1. Discriminant analysis

2.1.1. Fisher discriminant analysis

The objective of discriminant analysis is to find the features that best separate the different classes. One of the most used criterions $\mathscr{J}$ to reach is to maximize

$$\mathscr{J} = \mathrm{tr}(\boldsymbol{S}^E \boldsymbol{S}^I), \tag{1}$$

where the matrices $\boldsymbol{S}^E$ and $\boldsymbol{S}^I$ generally represent the scatter of sample vectors between different classes and within a class respectively. It has been shown (see Refs. [8,9]) that the $M \times D$ linear transform that satisfies

$$\hat{\boldsymbol{W}} = \arg \max_{\boldsymbol{W}^{\mathrm{T}} \boldsymbol{S}^I \boldsymbol{W} = \boldsymbol{I}} \mathrm{tr}(\boldsymbol{W}^{\mathrm{T}} \boldsymbol{S}^E \boldsymbol{W}) \tag{2}$$

optimizes the separability measure $\mathscr{J}$. This problem has an analytical solution based on the eigenvectors of the scatter matrices. The algorithm presented in Table 1 obtains this solution [9].

The most widely spread approach for defining the within and between class scatter matrices is the one that makes use of only up to second-order statistics of the data. This was proposed in a classic paper by Fisher [3] and the technique is referred to as FLD. In FLD the within class scatter matrix is usually computed as a weighted sum of the class-conditional sample covariance matrices. If equiprobable priors are assumed for classes C_k, $k = 1, \ldots, K$, then

$$\boldsymbol{S}^I = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{\Sigma}_k, \tag{3}$$

where $\boldsymbol{\Sigma}_k$ is the class-conditional covariance matrix, estimated from the sample set. The between class-scatter matrix is defined by

$$\boldsymbol{S}^E = \frac{1}{K} \sum_{k=1}^{K} (\boldsymbol{\mu}_k - \boldsymbol{\mu}_0)(\boldsymbol{\mu}_k - \boldsymbol{\mu}_0)^{\mathrm{T}}, \tag{4}$$

where $\boldsymbol{\mu}_k$ is the class-conditional sample mean and $\boldsymbol{\mu}_0$ is the unconditional (global) sample mean.

Notice the rank of $\boldsymbol{S}^E$ is $K-1$, so the number of extracted features is, at most, one less than the number of classes. Also notice the parametric nature of the scatter matrix. The solution provided by FLD is blind beyond second-order statistics, so we cannot expect this method to accurately indicate which features should be extracted to preserve any complex classification structure.

Table 1
General algorithm for solving the discriminability optimization problem stated in Eq. (2)

(1)	Given $\boldsymbol{X}$ the matrix containing data samples placed as N D-dimensional columns, $\boldsymbol{S}^I$ the within class scatter matrix, and M maximum dimension of discriminant space.
(2)	Compute eigenvectors and eigenvalues for $\boldsymbol{S}^I$. Make $\boldsymbol{\Phi}$ the matrix with the eigenvectors placed as columns and $\boldsymbol{\Lambda}$ the diagonal matrix with only the nonzero eigenvalues in the diagonal. M^I is the number of non-zero eigenvalues.
(3)	Whiten the data with respect to $\boldsymbol{S}^I$, to obtain M^I dimensional whitened data, $\boldsymbol{Z} = \boldsymbol{\Lambda}^{-1/2} \boldsymbol{\Phi}^{\mathrm{T}} \boldsymbol{X}$.
(4)	Compute $\boldsymbol{S}^E$ on the whitened data.
(5)	Compute eigenvectors and eigenvalues for $\boldsymbol{S}^E$ and make $\boldsymbol{\Psi}$ the matrix with the eigenvectors placed as columns and sorted by decreasing eigenvalue value.
(6)	Preserve only the first $M^E = \min\{M^I, M, \mathrm{rank}(\boldsymbol{S}^E)\}$ columns, $\boldsymbol{\Psi}_M = \{\psi_1, \ldots, \psi_{M^E}\}$ (those corresponding to the M^E largest eigenvalues).
(7)	The resulting optimal transformation is $\hat{\boldsymbol{W}} = \boldsymbol{\Psi}_M^{\mathrm{T}} \boldsymbol{\Lambda}^{-1/2} \boldsymbol{\Phi}^{\mathrm{T}}$ and the projected data, $\boldsymbol{Y} = \hat{\boldsymbol{W}} \boldsymbol{X} = \boldsymbol{\Psi}_M^{\mathrm{T}} \boldsymbol{Z}$.

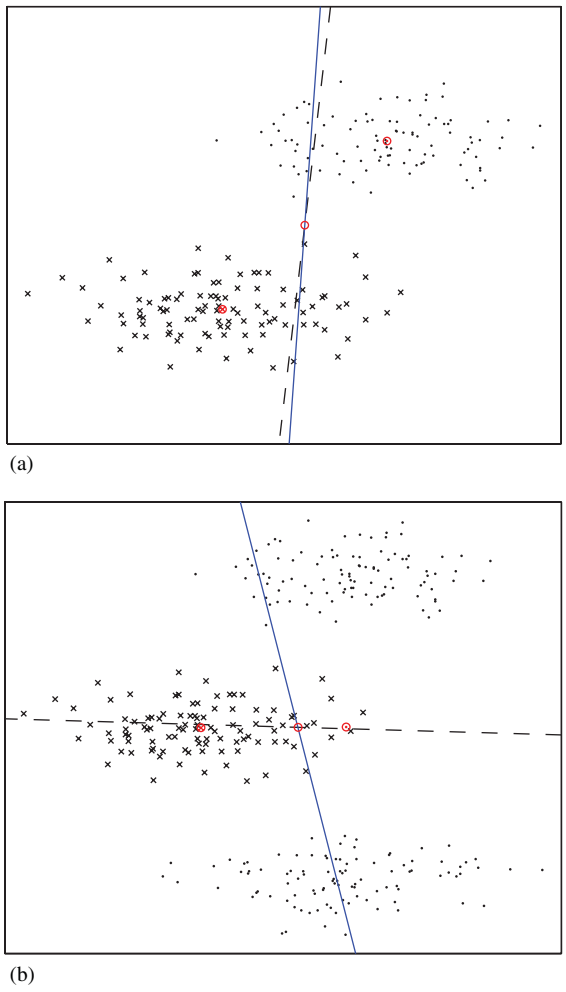

Fig. 1. First directions of NDA (solid line) and FLD (dashed line) projections, for two artificial datasets (a) and (b). Observe the results in (b), where the FLD assumptions are not met.

2.2. Nonparametric discriminant analysis

In Ref. [7] Fukunaga and Mantock present a nonparametric method for discriminant analysis in an attempt to overcome the limitations of FLD. In NDA the between-class scatter $\boldsymbol{S}^E$ is of nonparametric nature. This scatter matrix is generally full rank, thus loosening the bound on extracted feature dimensionality. Also, the nonparametric structure of this matrix inherently leads to extracted features that preserve relevant structures for classification. We briefly expose this technique, extensively detailed in Ref. [9].

In NDA, the between-class scatter matrix is obtained from vectors locally pointing to another class. This is done as follows. The extra-class nearest neighbor for a sample $\boldsymbol{x} \in C_k$ is defined by $\boldsymbol{x}^E = \{\boldsymbol{x}' \in \overline{C_k} / \|\boldsymbol{x}' - \boldsymbol{x}\| \leqslant \|\boldsymbol{z} - \boldsymbol{x}\|, \forall \boldsymbol{z} \in \overline{C_k}\}$. In the same fashion we can define the set of intra-class nearest neighbors by $\boldsymbol{x}^I = \{\boldsymbol{x}' \in L_c / \|\boldsymbol{x}' - \boldsymbol{x}\| \leqslant \|\boldsymbol{z} - \boldsymbol{x}\|, \forall \boldsymbol{z} \in C_k\}$.

From these neighbors, the extra-class differences are defined by $\boldsymbol{\Delta}^E = \boldsymbol{x} - \boldsymbol{x}^E$ and the intra-class differences by $\boldsymbol{\Delta}^I = \boldsymbol{x} - \boldsymbol{x}^I$. Notice that $\boldsymbol{\Delta}^E$ points locally to the nearest class (or classes) that does not contain the sample. The nonparametric between-class scatter matrix is defined by (assuming uniform priors),

$$\boldsymbol{S}^E = \frac{1}{N} \sum_{n=1}^{N} (\boldsymbol{\Delta}_n^E)(\boldsymbol{\Delta}_n^E)^{\mathrm{T}}, \tag{5}$$

where $\boldsymbol{\Delta}_n^E$ is the extra-class difference for sample $\boldsymbol{x}_n$.

A parametric form is chosen for the within-class scatter matrix $\boldsymbol{S}^I$, defined as in (3).

Fig. 1 illustrates the differences between NDA and FLD in two artificial datasets, one with Gaussian classes where results are similar, and one where FLD assumptions are not met. For the second case, the bimodality of one of the classes displaces the class mean introducing errors in the estimate of the parametric version of $\boldsymbol{S}^E$. The nonparametric version is not affected by this situation.

In Ref. [10] Bressan and Vitria introduced also a nonparametric form of the within-class scatter matrix $\boldsymbol{S}^I$, which is expected to improve the NN classification:

$$S_w = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\Delta}_n^I \boldsymbol{\Delta}_n^{I\mathrm{T}}. \tag{6}$$

In fact, the use of the nonparametric within scatter matrix achieves an intra-class normalization. Instead of assuming a gaussian distribution on the points of the same class, it normalizes the distances between each point and their nearest neighbors, which has been shown to benefit the nearest neighbor rule.

3. Discriminant embedding

Once the bases of discriminant analysis have been shown, we will focus on the construction of the embedding proposed in this work. We will not consider a linear projection built by maximizing some criteria over a statistic measure of the whole set of points. Instead of that, our mapping will take into account each point as a generator of a potential one-dimensional projection. Our objective is to build a high-dimensional discriminant embedding from a large set of potential one-dimensional discriminant projections. One of the simplest discriminant projections can be built considering that each point with its nearest neighbor of each class can generate a linear classifier, and the goal will be to find the embedding that optimally combines these classifiers minimizing the classification error.

Given a set of N points we build N simple classifiers as linear projections to a one-dimensional subspace. Then we choose the best combination of this projections to build the final embedding from the high-dimensional space to the lower one. Our approach uses the AdaBoost algorithm as a

method of classifier selection [11] and combination, which has been proved to be very efficient in machine learning literature. As we will show, we also obtain a way of ranking the axis of projection as a natural application of the classifier coefficients intrinsic of the boosting algorithm.

4. Obtaining discriminant 1D subspaces

Let $\boldsymbol{x_k}$ be a data point, $\boldsymbol{x_i}$ its nearest neighbor of the same class and $\boldsymbol{x_e}$ its nearest neighbor of the other class ($\boldsymbol{x_k}, \boldsymbol{x_i}, \boldsymbol{x_e} \in \boldsymbol{X}$). We will define the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ which point to $\boldsymbol{x_i}$ and $\boldsymbol{x_e}$ from $\boldsymbol{x_k}$.

We need to find a linear projection $f(\boldsymbol{x}) : X \to \mathbb{R}^d$ that minimizes the distance between the point $\boldsymbol{x_k}$ to the points of its same class, and maximizes the distance to the points of the other class. In our case we will deal with extreme dimensionality reduction ($d = 1$), so the projection matrix will be a simple vector. To find this vector we try to maximize the following criteria:

$$\boldsymbol{F_e}(\boldsymbol{\theta}) = \langle v, r\rangle - \langle u, r\rangle. \tag{7}$$

We try to find the direction $\boldsymbol{r}$ where the projection of the extra class vector is maximum and at the same time the projection of the intra class vector is minimum. The vector $\boldsymbol{r}$ is a generic rotation vector, defined as a function the unknown parameter θ:

$$\begin{pmatrix} \cos(\theta) - \sin(\theta) \\ \sin(\theta) + \cos(\theta) \end{pmatrix}. \tag{8}$$

We also have added a restriction to the problem, we impose that the vector solution lies in the plane defined by $\boldsymbol{u}$ and $\boldsymbol{v}$. So we project the points $\boldsymbol{x_k}, \boldsymbol{x_i}, \boldsymbol{x_e}$ into the plane, and only the 2-D problem must be solved. The resulting vectors $\boldsymbol{r}$ can be easily retroprojected to the original n-dimensional space. The 2-D problem has a closed solution by deriving F_e:

$$\begin{aligned} \frac{\partial F_E}{\partial \theta} = {} & \cos\,\theta(-u_1 + u_2 + v_1 - v_2) \\ & - \sin\,\theta(u_1 + u_2 - v_1 - v_2), \end{aligned} \tag{9}$$

$$\theta = \arctan\frac{-u_1 + u_2 + v_1 - v_2}{u_1 + u_2 - v_1 - v_2}. \tag{10}$$

Using the resulting θ it is straightforward to find the coordinates of the projection vector $\boldsymbol{r}$ in the n-dimensional space.

5. AdaBoost embedding

In this section the construction of the global embedding using the simple one-dimensional projections will be explained. We are interested in a combination of the one-dimensional projections that can yield a strong nearest neighbor classifier. Our scheme takes benefit of a very tested algorithm in machine learning to perform it: AdaBoost [12,13].

The use of boosting in our scheme is specially justified, because our 1D projections perform always as weak classifiers, and we can exploit the sample weight actualization intrinsic in the boosting scheme to focus the selection of the next feature axis to the examples more difficult to classify. This allows us to build a global embedding that combines simple discriminant projections that together can achieve more separability on the whole training data set. We also obtain a simple way of ranking the different features extracted, by taking into account the weights that receive each classifier at each step.

5.1. AdaBoost

We have followed a boosting implementation similar to the one proposed by Viola and Jones [14]. Given a training set of n points $X_{1\ldots n}$ (k points of the label 1 and m points of the label 2), the algorithm performs as follows:

1. First we define a set of weights $W_{1,1\ldots n}$ (each weight assigned to one vector). The weights corresponding to the class 1 are initialized to $\frac{1}{k}$, and the weights of the members of the class 2 are initialized to $\frac{1}{m}$.
   We also build the set of partial classifiers as 1D projections as it was defined in (10), so each sample X_i generates a projection to a 1D space. Notice that this projections are stored, and not recomputed again in the algorithm.
2. Then a fixed number of boosting steps are generated. At each boosting step s:
   - The whole set of classifiers is tested using the training points $X_{1\ldots n}$. We project each data point in the 1D space generated by each feature extraction and classify it according to its nearest neighbor. For each different projection, we evaluate its classification error by

$$Error_j = \sum_{i=1}^{n} W_{s,i} l_{i,j}, \tag{11}$$

     where $l_{i,j}$ is set to 1 if the point X_i has been correctly classified by the classifier j and to 0 otherwise. Finally we select the classifier c with minimum $Error_{1\ldots n}$.
   - Using the classification results of the classifier c, the set of weights W is actualized as:

$$W_{s+1,i} = W_{s,i}\beta^{1-l_{i,c}} \tag{12}$$

where

$$\beta = \frac{Error_c}{1 - Error_c}. \tag{13}$$

- The coefficient α_s corresponding to the classifier at the step s is computed as:

$$\alpha_s = \log\left(\frac{1}{\beta}\right). \tag{14}$$

- Finally the weights are normalized:

$$W_{s+1,i} = \frac{W_{s,i}}{\sum_j^n W_{s,j}}. \tag{15}$$

3. The output of the algorithm is a projection matrix, where we place at each column i_s the 1-D projection corresponding to the best classifier at the step s of the AdaBoost algorithm. In addition the $\alpha_{1\ldots s}$ coefficients are used to rank the importance of the features extracted for each 1-D projection.

The projection vectors have been selected taking into account the nearest neighbor classification rule of the training set, and at each step complementary projections are selected (trying to focus the projection on the misclassifed samples), so the final embedding should be a set of vectors that define an optimal subspace for classifying using the nearest neighbor rule.

The complexity of the learning algorithm is quadratic with respect to the number of training samples and the original dimensionality of the data. But it is important to note that the generation of the classifier pool is performed just once. Further learning can be done adding just the information of the new vectors. The complexity of the technique in operating time is reduced to a product matrix as is the case of the other discriminant analysis algorithms described.

6. Experiments

In order to see the performance of our embedding algorithm, we have tested it in two different experiments. First we have tried to solve a gender recognition problem, using the samples of two public available face databases, and then we have used some digits of the MNIST database [15] for the same purpose. In both cases we compare our scheme with classic discriminant analysis solutions.

6.1. Gender recognition

In this experiment we have taken a set of 2500 images from two public face databases: the AR face database [16] (leaving out the images with occlusions), and the XM2VTS [17]. There were 1323 male images and 1177 females in the global database. To perform our experiments we have broken the set of images into five independent sets (500 samples each set), and we have averaged the results of all our tests using a five-fold cross validation (using each time 500 images to train and 2000 to test).

Face images have been preprocessed before the gender recognition experiment. We have selected the center of the eyes, and we have aligned and cropped each face image according to the inter-eye distance (obtaining a 32×40 image) . Also we have normalized each image with respect to the mean and variance, in order to mitigate the effects of changes in illumination. In Fig. 2(a) we show some examples of the face images and their normalized versions. As can be seen only the internal features of the images have been taken into account, there is almost no hair or external information, what makes the problem harder. The final images are represented as a 1280-dimensional vector.

The results of our experiment show that using the embedding proposed we achieve significantly higher accuracy ratios than using any other discriminant projection. We have tested our embedding using two kind of distance measures. First we have used the nearest neighbor rule with Euclidean distance. Then we have used the L1 distance. Both algorithms achieve similar ratios. In the Fig. 3(a) we show the accuracies obtained as a function of the final dimension. We compare our embedding with the classic Fisher discriminant analysis (which is equivalent to a one-dimensional projection in a two class problem), and with the NDA algorithm using one and five nearest neighbors. We have selected the projection axis of our embedding according to their importance encoded in the coefficients $\alpha_{1\ldots s}$. The maximum classification rate is obtained using our embedding with Manhattan distance (89%).

6.2. MNIST digit database

In addition to the gender recognition problem, we have also built a completely different data set to test the performance of the embedding. We have selected two digits of the MNIST database to build a two class classification problem (we have selected the 1 and the 7 due to their similarity). Each digit is a 28×28 image that is represented as a 784-dimensional vector. As before we have tested the embedding with Euclidean and L1 distance, and also we have obtained similar results. Again the results obtained using our embedding are better than NDA and the nearest neighbor in the original space. The best accuracy is obtained using our embedding with euclidean distance (98%).

7. Conclusions

In this work we have proposed a new discriminant embedding that makes no assumptions on the data distribution and that shows better performance than related methods when used for data classification using the nearest neighbor rule. We have used the AdaBoost algorithm to select among a large set of simple projections that best separate the training

(a)

(b)

Fig. 2. Examples of the male and female images used in the experiments, taken from the AR face database, the XM2VTS and the MNIST. We also show the normalized version of the faces. (a) Original faces and their normalized version. (b) Some digits of the MNIST database.

data, so we have not restricted our projection to an orthogonal transformation. Moreover, our discriminability criteria has a nonparametric nature that does not require any global statistic measure on the input data. The resulting embedding is specially suited to the nearest neighbor rule by construction, a fact that is supported by empirical tests.

A notable weak point of the algorithm can be found when dealing with low-dimensional data. Further experiments made using the UCI repository with databases of dimensionality lower than 20 showed accuracies not significantly better than the ones obtained using classic FLD or NDA. So, the method we are proposing is specially suitable for feature extraction on high-dimensional subspaces, typical from computer vision problems.

There is still some future work to do. Further research about the influence of the weak 1D classifiers on the resulting embedding could be done. It is possible to extend this approach to the nonlinear case. Also other maximization criteria could be used to achieve higher performance or better learning rates. Also we plan as a feature work to

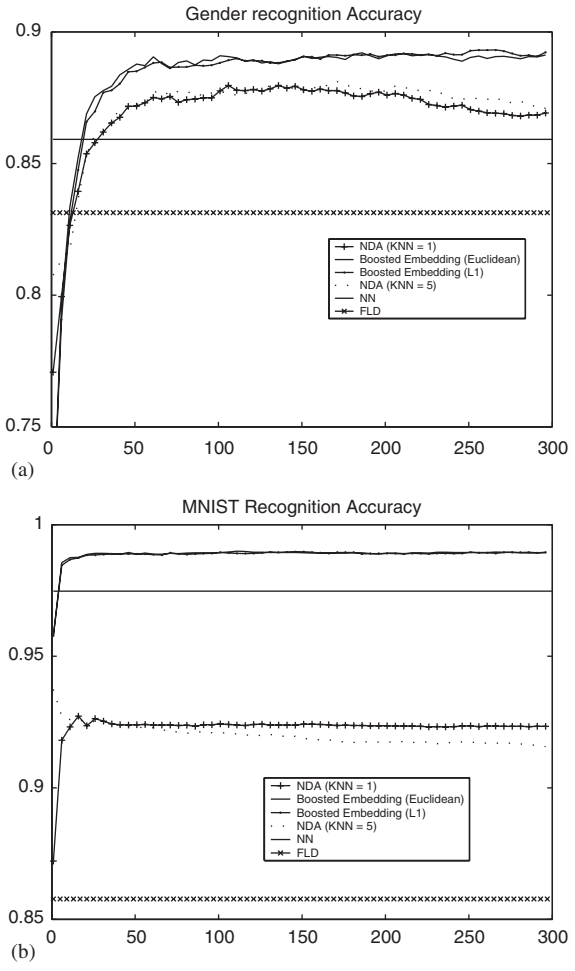

Fig. 3. Accuracies obtained in the Gender Recognition and the MNIST tests as a function of the dimensionality reduction (number of features extracted using each method). We also show the results using the nearest neighbor in the original space (which is shown as a horizontal line). (a) Gender recognition accuracies. (b) MNIST databases recognition accuracies.

extend the algorithm to the multi-class case. Two different approaches can be followed for this purpose: the most straightforward one is to use a pairwise based scheme using the 2-class algorithm. The second option is to extend a boosting algorithm specifically designed for the multi-class problem (Adaboost.M1).

References

[1] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cognitive Neurosci. 3 (1) (1991) 71–86.
[2] M. Kirby, L. Sirovich, Application of the Karhunen-Loeve procedure for the characterization of human faces, IEEE Trans. Pattern Anal. Machine Intell. 12 (1) (1990) 103–108.
[3] R. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugenics 7 (1936) 179–188.
[4] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (5500) (2000) 2319–2323.
[5] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.
[6] M. Aladjem, Linear discriminant analysis for two classes via removal of classification structure, IEEE Trans. Pattern Anal. Machine Intell. 19 (2) (1997) 187–192.
[7] K. Fukunaga, J. Mantock, Nonparametric discriminant analysis, IEEE Trans. Pattern Anal. Machine Intell. 5 (6) (1983) 671–678.
[8] P. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice-Hall, London, UK, 1982.
[9] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed., Academic Press, Boston, MA, 1990.
[10] M. Bressan, J. Vitria, Nonparametric discriminant analysis and nearest neighbor classification, Pattern Recognition Lett. 24 (15) (2003) 2743–2749.
[11] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: IEEE Conference on CVPR, Kauai, Hawaii, 2001, pp. 511–518.
[12] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: International Conference on Machine Learning, 1996, pp. 148–156.
[13] R.E. Schapire, A brief introduction to boosting, IJCAI, 1999, pp. 1401–1406.
[14] P. Viola, M. Jones, Robust real-time face detection, Int. J. Comput. Vision 57 (2) (2004) 137–154.
[15] Y. LeCun, The MNIST database of handwritten digits, URL http://yann.lecun.com/exdb/mnist/index.html
[16] A. Martinez, R. Benavente, The AR face database, Technical Report 24, Computer Vision Center (June 1998).
[17] J. Matas, M. Hamouz, K. Jonsson, J. Kittler, Y. Li, C. Kotropoulos, A. Tefas, I. Pitas, T. Tan, H. Yan, F. Smeraldi, J. Bigun, N. Capdevielle, W. Gerstner, S. Ben-Yacoub, Y. Abdeljaoued, E. Mayoraz, Comparison of face verification results on the xm2vts database, in: ICPR, 1999.