MULTI-CLASS DISCRIMINANT ANALYSIS BASED ON SVM ENSEMBLES FOR RANKING PRINCIPAL COMPONENTS

Tiene A. Filisbino

Diego Augusto T. Quadrado Leite

Gilson A. Giraldi

tiene@lncc.br

diegoamim@gmail.com

gilson@lncc.br

National Laboratory for Scientific Computing

25651-075, Rio de Janeiro, Petropolis, Brazil

Carlos E. Thomaz

cet@fei.edu.br

FEI, Department of Electrical Engineering

09850-901, São Paulo, São Bernardo do Campo, Brazil

Abstract. *The problem of ranking components obtained by subspace learning techniques is addressed by the discriminant principal components analysis (DPCA) that aims to identify the most discriminant subspaces using weights given by separating hyperplanes. Originally, the DPCA is a two-class discriminant feature technique. In this paper we extend the DPCA for multi-class problems. Specifically, we keep the principal component analysis (PCA) technique as a pre-processing step for dimensionality reduction. Then, given a N-class database, we build a linear support vector machine (SVM) ensemble, composed by N SVM machines, to get the discriminant weights that are combined through an AdaBoost technique in order to determine the discriminant contribution of each feature. To evaluate the Multi-Class DPCA components, we perform group separation tasks in facial expression experiments involving neutral, happiness, sad, anger, disgust, frontal face images. Our experimental results have shown that the principal components selected by the Multi-Class DPCA allow robust interpretation of the data, as well*

as higher recognition rates using less PCA features.

Keywords: *Discriminant Analysis, Principal Component Analysis, Support Vector Machine, Ensemble Methods, AdaBoost*

1 INTRODUCTION

Many areas such as pattern recognition, computer vision, signal processing and medical image analysis, require the managing of data sets with a large number of features or dimensions. Therefore, dimensionality reduction and discriminant analysis should be used for discarding redundancy and reduce the feature space for discriminating sample groups [Hastie et al. 2001].

For instance, classical works on linear dimensionality reduction including the classical principal component analysis (PCA), factor analysis (FA) [Fodor 2002], multidimensional scaling (MDS) [Hastie et al. 2001], projection pursuit (PP)[Safavi and Chang 2008, Fodor 2002] consider a gray scale $m_1 \times m_2$ image as a high dimensional vector in $\Re^{m_1 \cdot m_2}$ space. Linear techniques seek for new variables that obey some optimization criterium and can be expressed as linear combination of the original ones.

Once performed the dimensionality reduction, the next step in statistical learning includes the determination of discriminant features. This problem is very known in the context of principal component analysis (PCA). In this case, it was observed that, since PCA explains the covariance structure of all the data its most expressive components, that is, the first principal components with the largest eigenvalues, do not necessarily represent the most important discriminant directions to separate sample groups [Filisbino et al. 2013]. This observation motivates the application and development of other techniques to compute a discriminant subspace. These solutions depend on the definition of a criterium to quantify the importance of a given feature.

In general, Fisher's Linear Discriminant Analysis (LDA) [Devijver and Kittler 1982, Fukunaga 1990] is used to identify the most important linear directions for separating sample groups [Devijver and Kittler 1982, Fukunaga 1990] rather than PCA. This method, as well as the weighted pairwise variant of the well-known multi-class Fisher criterion introduced in [Loog et al. 2001] has the limitation of finding number of groups - 1 meaningful discriminant directions.

In [Thomaz and Giraldi 2010] it is proposed the discriminant principal components analysis (DPCA), based on the idea of using the discriminant weights obtained by separating hyperplanes to select among the principal components the most discriminant ones. So, by assuming that there are only two classes to separate the DPCA addresses the problem of selecting the discriminant principal components by using a linear classifier framework. The DPCA methodology presented in [Thomaz and Giraldi 2010] focuses on the LDA and linear support vectors machine (SVM) [Burges 1998, Vapnik 1998] methods although any other separating hyperplane could be used. The efficiency of the discriminant principal components selected by DPCA has been reported for two-group separation tasks in gender and facial expression experiments using frontal face images [Thomaz and Giraldi 2010].

In this paper we extend the DPCA for discriminant principal components selection in multi-class problems. The Multi-Class DPCA pipeline consists of the following steps: (a) apply PCA technique for dimensionality reduction in order to eliminate redundancy. (b) Compute a

linear SVM ensemble, based on the one-against-all SVM multi-class approach. (c) Combine the discriminant weights computed through the separating SVM hyperplanes in order to determine the discriminant contribution of each feature. So, given a N-class database, the step (b) builds N SVM machines in the PCA space. We implement the step (c) by adapting an ensemble technique, the AdaBoost one [Zhou 2012], to yield a global discriminant vector.

Ensemble methods find an accurate classifier by combining many moderately accurate learners [Zhou 2012]. The main loop of AdaBoost generates a collection of weak component classifiers and their corresponding AdaBoost weights. In the last step of the AdaBoost procedure, the component classifiers are linearly combined through the computed AdaBoost weights. This is the link between AdaBoost and our proposal.

Specifically, in each iteration of the AdaBoost procedure we build a linear SVM machine and its corresponding AdaBoost weight which composes the input for the final step of the ensemble method. Once we linearly combine linear classifiers to compute the strong classifier, it is straightforward to get the global discriminant weights from the expression that defines the strong learner. Then, following the traditional DPCA proposal, we sort PCA components in the decreasing order of the global discriminant weights. The method is not restricted to any particular probability density function of the sample groups because it can be based on either a parametric or non-parametric separating hyperplane approaches. However, it is known that a strong learner like SVM does not work well as the base learner for Adaboost. Therefore, we implement a strategy to compute a weakened version of SVM that is useful as an Adaboost component. To evaluate the Multi-Class DPCA algorithm, we perform group separation tasks in facial expression experiments involving neutral, happiness, sad, anger, disgust, frontal face images. The experiments show that the SVM can be used as an effective component classifier to generate the discriminant weights for the Multi-Class DPCA. Furthermore, the computational experiments demonstrate the benefits of sorting principal components using the Multi-Class DPCA if compared with the traditional PCA methodology of selecting as the principal components the ones with the largest eigenvalues.

The paper is organized as follows. The linear SVM classifier is presented in Section 2. Next, in Section 3, we review the theory behind PCA and DPCA. Section 4 presents the Multi-Class DPCA. The computational experiments are described in Section 5. Finally, in Section 6, we conclude the paper, summarizing its main contributions and describing further developments.

2 LINEAR SVM

SVM [Vapnik 1998] is primarily a two-class classifier that maximizes the width of the margin between classes, that is, the empty area around the separating hyperplane defined by the distance to the nearest training samples. As a consequence, SVM is more robust to outliers, zooming into the subtleties of group differences [Davatzikos 2004]. It can be extended to multi-class tasks by solving essentially several two-class problems [Yildizer et al. 2012]. Given a labeled training set:

$$X = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \ldots (\mathbf{x}_M, y_M)\}; \tag{1}$$

where $\mathbf{x}_i$ denote the n-dimensional training observations and $y_i \in \{-1, 1\}$ the corresponding classification labels, the SVM method [Vapnik 1998] seeks to find the hyperplane defined by

$$f(\mathbf{x}) = (\mathbf{x} \cdot \phi) + b = 0, \tag{2}$$

which separates positive and negative observations with the maximum margin. It can be shown that the solution vector ϕ_{svm} is defined in terms of a linear combination of the training observations, that is,

$$\phi_{svm} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i, \tag{3}$$

where α_i are non-negative coefficients obtained by solving a quadratic optimization problem with linear inequality constraints [Burges 1998, Vapnik 1998]. Those training observations $\mathbf{x}_i$ with non-zero α_i lie on the boundary of the margin and are called support vectors.

Behind the solution given by expression (3) is the assumption that there is a unit vector $\phi \in \Re^n$ and a constant $c \in \Re$ such that the inequalities:

$$\begin{cases} \mathbf{x}_i * \phi > c, \quad if \quad y_i = +1 \quad (a) \\ \mathbf{x}_j * \phi < c, \quad if \quad y_j = -1 \quad (b), \end{cases} \tag{4}$$

hold true ("$*$" denotes the usual inner product in $\Re^n$).

However, sometimes the classes may be nonseparables in the sense that we can not find an hyperplane $x * \phi = c$ such that conditions (4(a))-(4(b)) hold true. In this case, one solution is to work with a more convenient SVM formulation based on minimize the functional [Vapnik 1998]:

$$L = \frac{1}{2}(\phi * \phi) + C \sum_{i=1}^{n} \xi_i \tag{5}$$

subject to:

$$\begin{cases} \xi_i \geq 0, \quad i = 1, 2, ..., n, \\ y_i \left((\mathbf{x}_i * \phi) + b \right) \geq 1 - \xi_i, \quad i = 1, 2, ..., n., \end{cases} \tag{6}$$

Nonseparable SVMs allow the decision boundary to misclassify some examples. The constant C must be given in advance and controls the cost for the number of violated constraints. Once solved the optimization problem defined by expressions (5)-(6), the ϕ_{svm} solution is computed by expression (3).

3 PCA AND DISCRIMINANT ANALYSIS

One fundamental point in statistical learning is to find out an efficient feature space. In general the features are characterized by the knowledge of an human expert. However, once a feature space has been defined, a key question is "how can we determine (or compute) the most important discriminant features without loosing important information for pattern recognition

tasks, like classification?" Discriminant analysis techniques address this question, which is very known in the context of PCA.

In this case, it was observed that, since PCA explains the covariance structure of all the data its most expressive components [Swets and Weng 1996], that is, the first principal components with the largest eigenvalues, do not necessarily represent important discriminant directions to separate sample groups. The Figure 1 is a simple example that pictures this fact. Both Figures 1.(a) and 1.(b) represent the same data set. Figure 1.(a) just shows the PCA directions ($\widetilde{x}$ and $\widetilde{y}$) and the distribution of the samples over the space. However, in Figure 1.(b) we distinguish two patterns: plus $(+)$ and triangle $(\blacktriangledown)$. We observe that the principal PCA direction $\widetilde{x}$ can not discriminate samples of the considered groups.

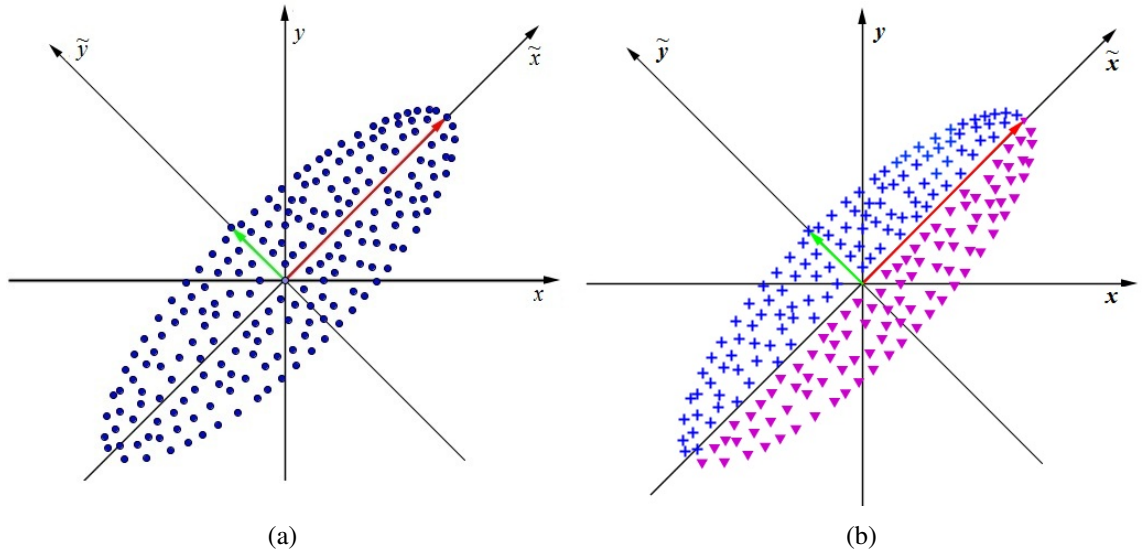


Figure 1: (a) Scatter plot and PCA directions. (b) The same population but distinguishing patterns plus $(+)$ and triangle $(\blacktriangledown)$.

This observation motivates the application and development of other techniques to compute a discriminant subspace. This problem depends on the definition of a criterium to quantify the importance of a given feature. Besides, we must be aware about the fact that this process is application dependent. Next, we review the PCA and DPCA approaches which are fundamental for this work.

3.1 Principal Components Analysis (PCA)

PCA is a feature extraction procedure concerned with explaining the covariance structure of a set of variables through a small number of linear combinations of these variables. Let an $M \times n$ training set matrix $\widetilde{\Theta}$ be composed of M input samples (or face images) with n variables (or pixels) centered respect to the global mean, computed by:

$$\widehat{\mathbf{x}} = \frac{1}{M} \sum_{i=1}^{M} \mathbf{x}_i \tag{7}$$

This means that each column of matrix $\widetilde{\Theta}$ represents the values of a particular variable observed all over the M samples centered respect to the corresponding mean. Let this data matrix

$\widetilde{\Theta}$ have covariance matrix Ω with respectively P and Λ eigenvector and eigenvalue matrices, that is:

$$P^T \Omega P = \Lambda. \tag{8}$$

It is a proven result that the set of m' ($m' \leq n$) eigenvectors of Ω, which corresponds to the m' largest eigenvalues, minimizes the mean square reconstruction error over all choices of m' orthonormal basis vectors [Fukunaga 1990]. Such a set of eigenvectors that defines a new uncorrelated coordinate system for the training set matrix $\widetilde{\Theta}$ is known as the principal components. In the context of face recognition, those $P_{pca} = [p_1, p_2, ..., p_{m'}]$ components are frequently called eigenfaces [Turk and Pentland 1991]. The PCA methodology is summarized in the Algorithm 1.

Algorithm 1: Procedure to compute PCA matrix.

1: Compute the global mean $\widehat{\mathbf{x}}$ of the input data through expression (7);
2: Centering input samples: $\widetilde{\mathbf{x}}_i = \mathbf{x}_i - \widehat{\mathbf{x}}$;
3: Build the centered data matrix $\widetilde{\Theta} = [\widetilde{\mathbf{x}}_1^T \widetilde{\mathbf{x}}_2^T \cdots \widetilde{\mathbf{x}}_M^T]$;
4: Determine $P_{pca} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{m'}]$.

3.2 Two-Class DPCA

The original DPCA is implemented taking as input a training set X, like in expression (1), to construct both LDA and SVM separating hyperplanes, although any other linear classifier could be used [Thomaz and Giraldi 2010]. Firstly, each data vector is centered respect to the mean, like in the PCA procedure described in the Algorithm 1. For discarding redundancies, the PCA transformation matrix $P_{pca} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{m'}]$ is computed by retaining the m' principal components with non-zero eigenvalues. Then, each zero mean data vector $\widetilde{\mathbf{x}}_i$ is projected on the principal components and reduced to a m'-dimensional vector $\widetilde{\mathbf{v}}_i$ representing the PCA features of each one of the m-dimensional input data. Afterwards, the $M \times m'$ data matrix and their corresponding labels are used as input to calculate the separating hyperplanes.

Since DPCA assumes only two classes to separate, there are only one discriminant vector, given by expressions (3) if we apply the SVM hyperplane. The most discriminant feature of each one of the m'-dimensional vectors $\widetilde{\mathbf{v}}_i$ is obtained by multiplying the $M \times m'$ most expressive features matrix by the $m' \times 1$ discriminant SVM vector. Thus, the initial training set consisting of M measurements on n variables is reduced to a data set consisting of M measurements on only 1 most discriminant feature given by:

$$\begin{aligned} c_1 &= \widetilde{v}_{11} w_1 + \widetilde{v}_{12} w_2 + ... + \widetilde{v}_{1m'} w_{m'}, \\ c_2 &= \widetilde{v}_{21} w_1 + \widetilde{v}_{22} w_2 + ... + \widetilde{v}_{2m'} w_{m'}, \\ &... \\ c_M &= \widetilde{v}_{N1} w_1 + \widetilde{v}_{N2} w_2 + ... + \widetilde{v}_{Nm'} w_{m'}, \end{aligned} \tag{9}$$

where $[w_1, w_2, ..., w_{m'}]^T$ are the weights corresponding to the calculated SVM separating hyperplanes on the principal PCA subspace.

Once the classifier has been constructed, we can determine the discriminant contribution of each feature by investigating the weights $[w_1, w_2, ..., w_{m'}]$ of the respective most discriminant directions. Weights that are estimated to be 0 or approximately 0 have negligible contribution on the discriminant scores c_i described in equation (9), indicating that the corresponding features are not significant to separate the sample groups. In contrast, largest weights (in absolute values) indicate that the corresponding features contribute more to the discriminant score and consequently are important to characterize the differences between the groups.

Therefore, instead of sorting these features by selecting the corresponding principal components in decreasing order of eigenvalues, as PCA does, DPCA selects as the most important features for classification the ones with the highest discriminant weights, that is, $|w_1| \geq |w_2| \geq ... \geq |w_{m'}|$. The subset of features with $w_i \neq 0$ are called the discriminant ones and the features with the highest weights are the most discriminant ones. Certainly, these highest weights might vary depending on the separating hyperplane used. Also, we can bypass the PCA and apply this approach to the original feature space. However, for high dimensional spaces, like in the image analysis applications, the result may be doubtful due to redundancies in the original image representation. The whole DPCA methodology can be summarized as follows.

Algorithm 2: Procedure for DPCA.

1: Determine P_{pca} according to the Algorithm 1
2: Form the matrix $\overline{\Theta}$ given by $\overline{\Theta} = (P_{pca})^T \widetilde{\Theta}^T$.
3: Compute ϕ_{svm} using expressions (3), with input data given by the columns of $\overline{\Theta}$ above.
4: Sort the discriminant weights such that $|w_1| \geq |w_2| \geq ... \geq |w_{m'}|$.
5: Select the principal components according to the obtained $|w_i|$ sequence.

4 MULTI-CLASS DISCRIMINANT ANALYSIS

Our proposal for the Multi-Class DPCA technique is based on the weakened SVM proposed in [Garcia and Lozano 2007], the AdaBoost algorithm described in [Zhou 2012], page 25, the AdaBoost.MH methodology presented in [Schapire and Singer 1999], and the nonseparable linear SVM computed by expressions (3) and (5)-(6). Our algorithm belongs to the class of boosting procedures, which refers to a family of algorithms that are able to build strong learners through weak learners as the basic components. Boosting procedures demonstrate that the complexity class of weakly learnable and strongly learnable problems are equal, since with these procedures any weak learner is potentially able to be boosted to a strong learner [Zhou 2012].

However, our basic learner in this paper is the SVM which can not be considered a weak learner. In fact strong learners have shown performance degradation in boosting algorithms [Li et al. 2008]. In order to address this problem and to avoid the presence of over-weak SVM or over-strong SVM component classifiers in the boosting process some strategies have been proposed in the literature. For instance, in [Li et al. 2008] it is described the AdaBoostSVM algorithm which uses a sequence of trained SVM, with RBF kernel (RBFSVM), as component classifiers. The key idea is to progressively adjust the RBFSVM control parameter σ starting with a large σ value which is reduced during the boosting iterations. In this way, the AdaBoostSVM starts with a weak learner which is improved as the boosting iteration proceeds. Behind the solution proposed in [Li et al. 2008] there is the idea of taking advantage of strong learners

in boosting methods by using a strategy to get a weakened version of the classifier. Hence, in [Garcia and Lozano 2007] it is proposed another methodology to build a weak SVM, which discards a percentage μ of the samples in the original data set to generate the training set. The Algorithm 3 summarizes this strategy which is applied in the extension of the DPCA algorithm for multi-class problems.

Algorithm 3: WSVM Procedure: Build a Weakened version of SVM.

Input: Labeled samples:$X = \{(\mathbf{x}_i, y_i), i = 1, 2, \cdots, n'\}$ where $y_i \in Y$ is the label of the sample $\mathbf{x}_i$;
Samples probability distribution $D(\mathbf{x}_i)$;
Percentage μ;
Select $\mathcal{J}$ so that, $\sum_{j \in \mathcal{J}} D(\mathbf{x}_j) \leq (1 - \mu)$;
Select $(x_i, y_i); i \in \mathcal{J}$, and define $D^* = D_{\mathcal{J}}$;
Compute the weighted data $X^* = \{(D_i^* \cdot \mathbf{x}_i, y_i), i \in \mathcal{J}\}$
Compute the (weak) SVM hyperplane ϕ_{svm} using X^* in expression (3);
Output: WSVM hyperplane ϕ_{svm}.

The Multi-Class DPCA methodology is described by the Algorithm 4. The training instances in the input database X are supposed independently and identically distributed from an uniform distribution D_1, at the initialization of the pipeline (line 1 of the Algorithm 4). Multi-Class DPCA also applies the PCA, revised on section 3.1, in its first stage. Like in the binary DPCA, the goal of this step is dimensionality reduction in order to eliminate redundancy. Next, the Multi-Class DPCA computes a SVM ensemble, based on the one-against-all SVM multi-class approach presented in [Yildizer et al. 2012]. So, let N be the total number of classes. Each iteration t of the Algorithm 4 constructs one weakened SVM (line 7 of Algorithm 4), in the PCA subspace, using the Algorithm 3. So, in line 6 of Algorithm 4, we take all k_t images from class t and label them as 1. Then, using random sampling we choose $(2k_t)/(N-1)$ images from classes other than t and label them as -1. The obtained set of feature vectors $\mathbf{x}_m^t$ and corresponding labels l_m:

$$X^t = \left\{ \left(\mathbf{x}_1^t, l_1\right), \left(\mathbf{x}_2^t, l_2\right), ... \left(\mathbf{x}_{3k_t}^t, l_{3k_t}\right), \quad \mathbf{x}_m^t \in \Re^n, \quad l_m \in \{-1, 1\} \right\}, \tag{10}$$

are the input to construct the WSVM model t by the Algorithm 3.

The lines (8)-(12) of the Algorithm 4 are based on the idea of deriving AdaBoost method by using the linear combination of weak (SVM) learners $h_1, h_2, \cdots, h_N$:

$$H(\mathbf{x}) = \sum_{t=1}^{N} \alpha_t h_t(\mathbf{x}), \tag{11}$$

where α_t does not depend on $\mathbf{x}$ [Zhou 2012]. In this formulation, we seek a strong learner H that minimizes the exponential loss function:

$$l_{exp}(H|D) = E_{\mathbf{x} \sim D}\left[e^{-f(\mathbf{x})H(\mathbf{x})}\right] = \sum_i e^{-f(x_i)H(x_i)} D(x_i),$$

where f is the ground-truth label function and D is a probability distribution [Zhou 2012].

The exponential loss is used since it gives an elegant and simple update formula, and it is consistent with the goal of minimizing classification error and can be justified by its relationship to the standard log likelihood. For simplicity, consider binary classification on classes with label set $\{-1, +1\}$.

The strong learner H is produced by iteratively generating h_t and α_t. The first weak classifier h_1 is generated by invoking the weak learning algorithm on the original distribution. When a classifier h_t is generated under the distribution D_t, its coefficient α_t in expression (11), named AdaBoost weight in this paper, is to be determined in order to minimizes the exponential loss:

$$l_{exp}\left(\alpha_t h_t | D_t\right) = E_{\mathbf{x}\sim D_t}\left[e^{-f(\mathbf{x})\alpha_t h_t(\mathbf{x})}\right] = \sum_i e^{-f(x_i)\alpha_t h_t(x_i)} D_t\left(x_i\right) =$$

$$\sum_{i;f(x_i)=h_t(x_i)} e^{-\alpha_t} D_t\left(f\left(x_i\right) = h_t\left(x_i\right)\right) + \sum_{i;f(x_i)\neq h_t(x_i)} e^{\alpha_t} D_t\left(f\left(x_i\right) \neq h_t\left(x_i\right)\right) =$$

$$e^{-\alpha_t}\left(1-\epsilon_t\right) + e^{\alpha_t}\epsilon_t,$$

where:

$$\epsilon_t = \left[\sum_{i;f(x_i)\neq h_t(x_i)} D_t\left(f\left(x_i\right) \neq h_t\left(x_i\right)\right)\right].$$

To obtain the optimal α_t, we must solve the equation:

$$\frac{\partial l_{exp}\left(\alpha_t h_t | D_t\right)}{\partial \alpha_t} = -e^{-\alpha_t}\left(1-\epsilon_t\right) + e^{\alpha_t}\epsilon_t = 0, \tag{12}$$

whose solution, after a simple algebra, is given by:

$$\alpha_t = \frac{1}{2} ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{13}$$

Once a sequence of weak classifiers and their corresponding AdaBoost weights have been generated, these classifiers are linearly combined to build H_{t-1}. The sample distribution D_t is updated such that in the next iteration the AdaBoost algorithm outputs a weak classifier h_t that corrects some mistakes of H_{t-1}. The mathematical formulation of this idea is obtained by minimizing the corresponding exponential loss function and it gives rise to the expression in line 12 of the Algorithm 4 (see [Zhou 2012] for details).

The final step of the Multi-Class DPCA procedure is justified by expression (11). In fact if $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_N$ are the solution vectors (equation (3)) that defines the weak linear SVM classifiers then:

$$H\left(\mathbf{x}\right) = \sum_{t=1}^{N} \alpha_t h_t\left(\mathbf{x}\right) = \sum_{t=1}^{N} \alpha_t < \mathbf{w}_t, \mathbf{x} > \equiv < \mathbf{w}_{mdpca}, \mathbf{x} >,$$

where:

$$\mathbf{w}_{mdpca} = \sum_{t=1}^{N} \alpha_t \mathbf{w}_t, \tag{14}$$

defines the global discriminant weights. Therefore, the manner that AdaBoost combines the weak classifiers gives a straightforward way to combine the discriminant weights, computed through the separating WSVM hyperplanes, by using expression (14). In order to determine the discriminant contribution of each feature, it is just a matter of sorting PCA components in decreasing order of the absolute value of the global weights defined by expression (14). The obtained discriminant weights are computed in line 13 of the Algorithm 4. The output of the Multi-class DPCA procedure is the discriminant principal components $\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{m'}$, where $\mathbf{q}_i$ is a PCA component ordered according to its discriminant weight.

Algorithm 4: Multi-Class DPCA procedure

Input:
Samples: $X = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \ldots (\mathbf{x}_M, y_M)\}$; where $y_i \in Y$ and $Y = \{1, 2, 3, .., N\}$;
Percentage μ;

1 Initialize the homogeneous distribution $D_1(\mathbf{x}_i) = \frac{1}{M}$, $i = 1, 2, \cdots, M$;
2 Determine P_{pca} following the Algorithm 1
3 Calculate the projected data $\overline{\mathbf{x}}_i = (P_{pca})^T \widetilde{\mathbf{x}}_i$
4 Build the labeled projected data set $\overline{\Theta} = \{(\overline{\mathbf{x}}_1, y_1), (\overline{\mathbf{x}}_2, y_2) \ldots (\overline{\mathbf{x}}_M, y_M)\}$
5 **for** $t = 1, ...$ ***to*** N **do**
6 Build the subset $\overline{\Theta}^t$, by taking all k_t projected images from class t and label them as $+1$. Use random sampling to choose $\frac{2k_t}{(N-1)}$ images from classes other than t and label them as -1;
7 Call the procedure $WSVM(\overline{\Theta}^t, \mathcal{Y}, D_t, \mu)$ where $\mathcal{Y} = \{-1, 1\}$
8 Compute: $e_t = \sum_{\forall i | y_i \neq h_t(\overline{\mathbf{x}}_i)} D_t(x_i)$ for all $\overline{\mathbf{x}}_i \in \overline{\Theta}^t$;
9 **if** $e_t > 0.5$ **then**
10 break;
11 Calculate AdaBoost weight: $\alpha_t = \frac{1}{2}\left(\frac{1-e_t}{e_t}\right)$;
12 Update: $D_{t+1}(i, l) = \frac{D_t(i,l) exp(-\alpha_t y_i [l] h_t(\widetilde{\mathbf{x}}_i, l))}{Z_t}$, where Z_t is a normalization factor which enables D_{t+1} to be a distribution.
13 Compute discriminant weights: $|w_{mdpca,i}| = |\sum_{t=1}^{N} \alpha_t w_{t,i}|$;
14 Sort the discriminant weights: $|w_{mdpca,1}| \geq |w_{mdpca,2}| \geq \cdots \geq |w_{mdpca,m'}|$;
15 Select the principal components according to the obtained $|w_{mdpca,i}|$ sequence;

Output: Discriminant principal components: $\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{m'}$

5 COMPUTATIONAL EXPERIMENTS

In this section we perform facial expression experiments using the frontal poses of the Radboud (RaFD) face image database which is an initiative of the Behavioural Science Institute of the Radboud University Nijmegen [Langner et al. 2010]. It was generated with 39 Caucasian Dutch adults, being 19 female and 20 male, displaying 8 emotional expressions with 5 colorful images with resolution of 1024×681 pixels, taken from five different poses in highly controlled environment. The images in Figure 2 show the neutral, happiness, sad, disgust, and anger frontal profile of a person that have been used in the experiments. In order to save memory allocation along the algorithms execution, we convert each pose to gray scale and resize it to

50×50 before computation. We can notice that the images in Figure 2 are well scaled and aligned. These features make this database very attractive for testing dimensionality reduction and discriminant analysis techniques.

(a) (b) (c) (d) (e)

Figure 2: Images from RaFD database. (a) Sample of neutral class. (b) Image of happiness class. (c) Sample of sad class. (d) Sample of disgust class. (e) Smaple of anger class.

Table 1, lists the 20 principal components with the highest discriminant weights given by Multi-Class DPCA (Algorithm 4), in absolute values, for discriminating the expression samples, where each line gives also the number of classes used in the tests:

- **Three-Class** experiment: neutral, happiness, and sad samples;
- **Four-Class** experiment: neutral, happiness, sad, and disgust images;
- **Five-Class** experiment: neutral, happiness, sad, disgust, and anger classes.

We can observe that the Multi-Class DPCA has selected some distant PCA components among the first 20 most discriminant principal components, such as the $43th$ one for the three-class experiment, the $59th$ and $49th$ ones for the four-class and five-class tests, respectively. Besides, when comparing the three lines of Table 1 we notice that the first three most discriminant principal components selected by Multi-Class DPCA in line 1 are more distant PCA components than in the line 2 and 3. We expect some consequences of this fact in the classification experiments, as we will see next.

Expression Experiment: PCA components sorted by the MultiClass DPCA procedure.																					
1	3 classes	24	26	20	9	14	12	7	5	8	22	4	36	21	43	25	17	32	1	29	23
2	4 classes	2	1	4	13	24	15	10	22	16	27	35	9	8	5	28	59	36	18	19	40
3	5 classes	2	1	6	3	16	10	18	32	30	7	37	21	36	49	31	33	44	17	47	13

Table 1: Top 20 (from top to bottom and left to right) discriminant principal components, ranked by Multi-Class DPCA, using the Radboud database

To understand the changes described by the principal components, we reconstruct the most expressive features by varying each principal component $\mathbf{p}_i$ separately using the equation:

$$I = \widehat{\mathbf{x}} + \beta \cdot \mathbf{p}_i, \tag{15}$$

where $\widehat{\mathbf{x}}$ is given by expression (7), $\beta \in \{\pm j \cdot \bar{\lambda}^{0.5}, \quad j = 0, \pm 3\}$, and $\bar{\lambda}$ is the average eigenvalue of the total covariance matrix S. We choose $\bar{\lambda}$ instead of λ_i because some λ_i can be very small (or big) in this case, showing no changes (or color saturation) between the samples when we move along the corresponding principal components.

Figure 3 illustrates the transformations on the first PCA most expressive component contrasted with the first discriminant principal component selected by the Multi-Class DPCA to separate facial expressions.

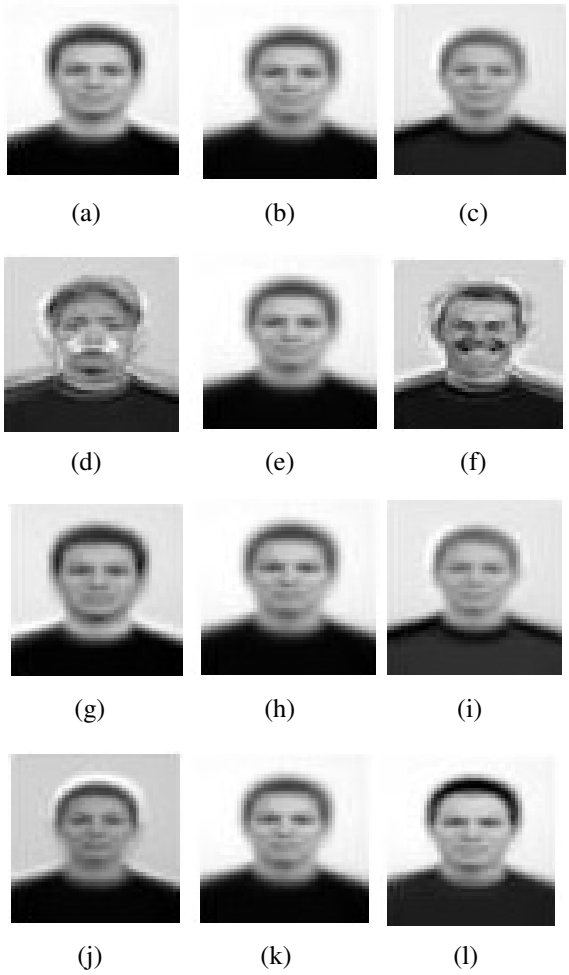


Figure 3: Visualization of the changes described by the principal directions using expression (15). From top to bottom: the first PCA most expressive component for the three-class task; the first Multi-Class DPCA most discriminant component for three-class test; the first PCA most expressive component for five-class experiment; and the first Multi-Class DPCA most discriminant principal component for five-class experiment.

In figure 3, it can be seen that the first PCA most expressive direction captures essentially the changes in gender, which are the major variations of all the training samples. Owing to the fact that changes in facial expression are much less significant than the gender ones, the standard PCA is unable to capture such minor variations in its first most expressive component. However, when we compare these results with the ones reconstructed by the Multi-Class DPCA most discriminant principal component, illustrated by Figures 3.(d)-(f), we can see that other principal component (the 24 according to Table 1) carries more information about expression variations then the first PCA one. Therefore, when PCA components are ranked by their largest discriminant weights rather than eigenvalues the result may be more effective with respect to extracting group-differences information. However, the superiority of Multi-Class DPCA principal component is not so evident when comparing the third row (Figures 3.(g)-(i)) with the

bottom row. Both methods seems to capture mainly changes in gender. This result suggest that when increasing the number of classes the performance of Multi-Class DPCA for separating samples groups decreases, as we will see next.

The following recognition tasks experiments are carried out using the full rank PCA subspace (Algorithm 1) with all non-zero eigenvalues. We use the 10-fold cross validation method to evaluate the classification performance of the Multi-Class DPCA, described by the Algorithm 4, in the PCA subspaces. In these experiments we have assumed equal prior probabilities and misclassification costs for all the class. On the PCA subspace, the mean of each class i has been calculated from the corresponding training images and the Mahalanobis distance from each class mean $\widehat{\mathbf{x}}_i$ has been used to assign a test observation $\mathbf{x}_r$ to either the different facial expressions. That is, we have assigned $\mathbf{x}_r$ to class i that minimizes:

$$d_i(\mathbf{x}_r) = \sum_{j=1}^{k} \frac{1}{\lambda_j}(x_{rj} - \widehat{x}_{ij})^2, \tag{16}$$

where λ_j is the corresponding eigenvalue, k is the number of principal components retained, x_{rj} and $\widehat{x}_{ij}$ are the projections of the sample $\mathbf{x}_r$ and of the mean $\widehat{\mathbf{x}}_i$, respectively, in the jth component considered.

The Figure 4 shows the average recognition rates of the 10-fold cross validation experiments for PCA and Multi-class DPCA, with the latter being referenced by the acronym **MD-PCA**. We can notice that for the three-class recognition task the Multi-Class DPCA method achieves higher recognition rates than the traditional PCA when considering $k < 60$. For $60 \leq k \leq 80$ the recognition rates of Multi-Class DPCA are higher or equal to the ones for PCA subspaces. In the range $81 \leq k \leq 100$, the PCA method outperforms the Multi-Class DPCA. We observe that the recognition rates plot of the discriminant components given by the largest Multi-Class DPCA weights achieves the maximum of 83% in $k = 50$ while for PCA method, the maximum is 80% achieved in $k = 60$. For 4 and 5 classes we notice a decreasing in the recognition rates for both Multi-Class DPCA and PCA. The results pictured on Figure 3 already indicated this fact. However, the Multi-Class DPCA outperforms the PCA almost everywhere for the four and five-class experiments in the range $k > 1$. Besides, in general, the recognition rates of Multi-Class DPCA components are higher for four-class than for five-class tests.

In order to get a visual insight about the differences between Multi-Class DPCA and PCA principal components, let us consider the recognition rates for the subspace generated by the first three principal components selected by these techniques for the three-class experiment. In Figure 4, the Multi-Class DPCA method shows a recognition rate above 50% for this subspace while PCA method achieve 36% in this case. The superiority of the Multi-Class DPCA subspaces with fewer dimensions can be visualized in Figure 5 which shows the samples projected on the first three principal components select by PCA and Multi-Class DPCA methods.

A visual inspection of Figures 5.(a) shows that PCA fails to recover the important features for separating the classes if compared with the Multi-Class DPCA result pictured on Figure 5.(b). This observation agrees with the recognition rates reported for Multi-Class DPCA on

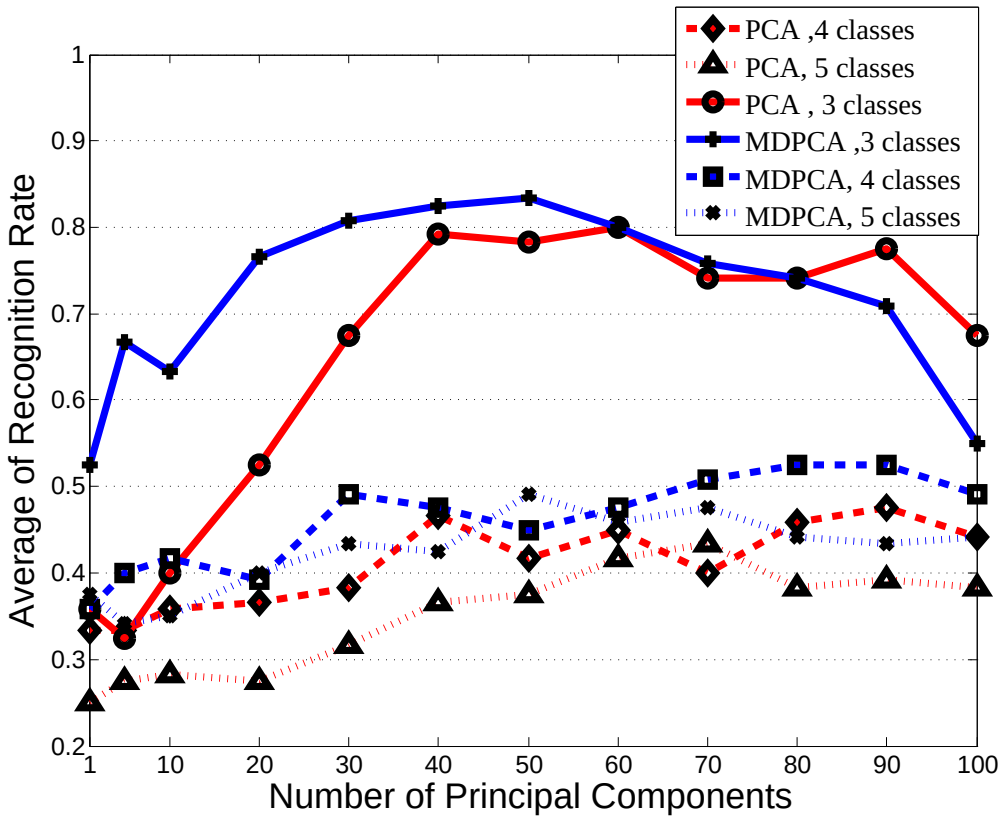


Figure 4: Average recognition rate of PCA selected by the largest eigenvalues and largest Multi-Class DPCA (MDPCA) discriminant weights criteria.

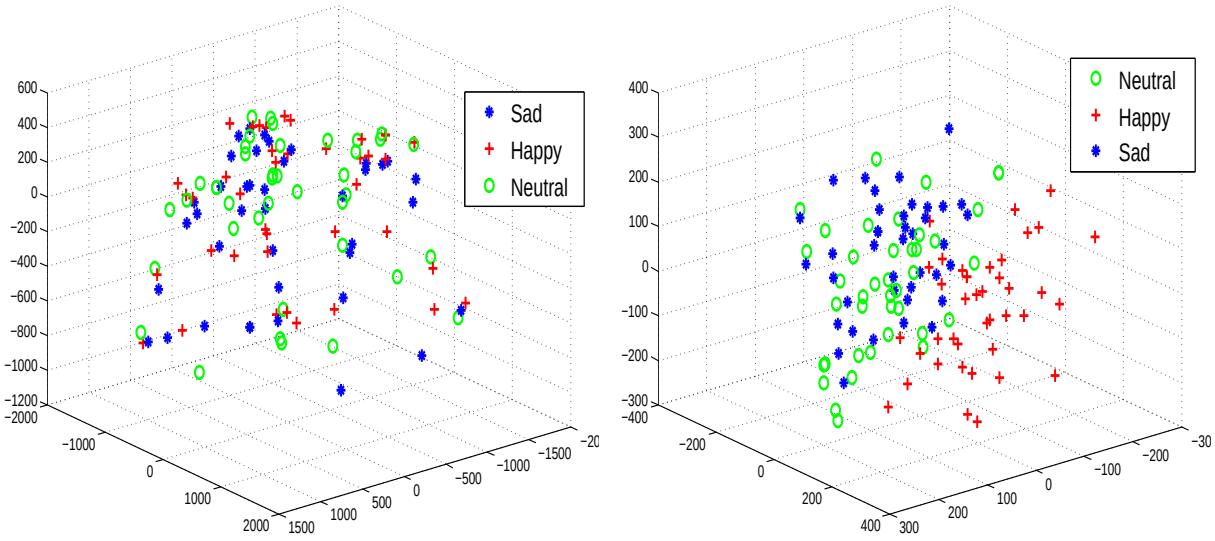
Figure 4 if compared with the recognition rates for PCA. In facial expression experiments, the feature that most vary in the samples (gender) are not necessarily the most discriminant ones. That is the point behind the above result. This fact explains also the degradation in the Multi-Class DPCA subspaces for the four and five-class experiments in the range $1 \leq k \leq 3$ because Multi-Class DPCA discriminant components are closer to PCA ones in this k interval (see Table 1). Consequently, they are more related to gender than to facial expressions. That is a problem to be considered in further works.

Now, it is worthwhile to consider the accumulated variance explained by the selected subspaces which is computed by:

$$Variance_{acc}^{l,i}(k) = \frac{\sum_{j=1}^{k} \lambda_j^l}{\sum_{j=1}^{m'} \lambda_j^l}, \tag{17}$$

where $l \in \{PCA, MDPCA\}$, $i \in \{3, 4, 5\}$, and λ_j^l is the variance associated to the jth component selected by the $l = PCA$ or $l = MDPCA$. The expression (17) is important for this discussions because we expect some correlation between the performance for reconstruction and the accumulated variance in expression (17) due to the already known fact that the components with larger variances keep global information related to features that most vary in the samples [Filisbino et al. 2013]. In Figure 6 we plot the result of expression (17).

We notice that, in general, the PCA technique gives the largest accumulated variances although its recognition rates are, in general, outperformed by the Multi-class DPCA components.



(a) (b)

Figure 5: Three-dimensional most expressive subspaces, for 3 class experiment, identified by PCA and Multi-Class DPCA. (a) Projection of samples in the PCA subspace. (b) Samples projected in the Multi-Class DPCA subspace.

As already discussed in section 3, since PCA explains features that most vary in the samples the principal subspaces do not necessarily represent important discriminant directions to separate sample groups. However, the reconstruction results are expected to give lower errors if we take components with higher variances. To make clear this observation, let us quantify the reconstruction quality through the root mean squared error (RMSE), computed as follows:

$$RMSE^{l,i}(k) = \sqrt{\frac{\sum_{i=1}^{N} ||P.I_k^l.P^T x_i - x_i||^2}{N}}, \tag{18}$$

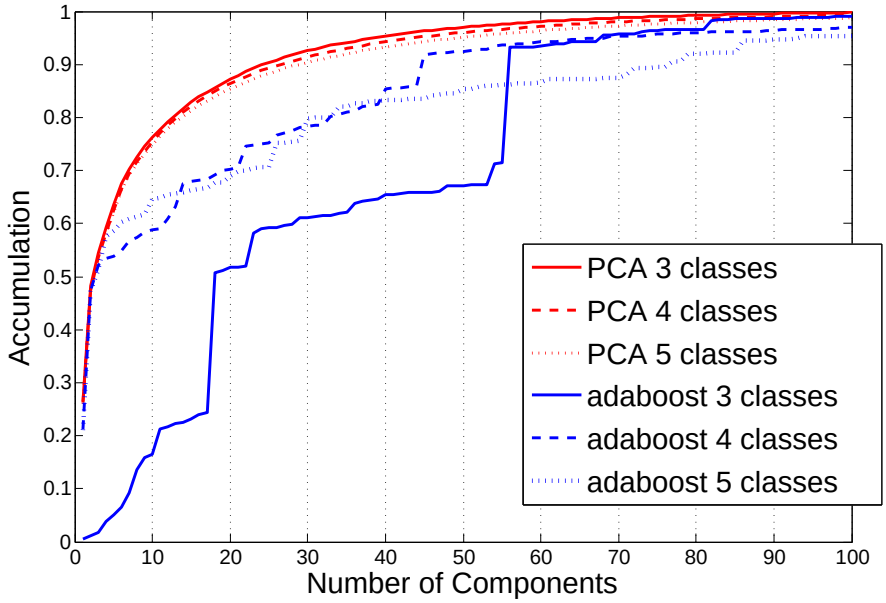
where $l \in \{PCA, MDPCA\}$, I_k^l is a truncated identity matrix that keeps the subspace with dimension k that is selected by PCA or Multi-Class DPCA, and $\|\cdot\|$ is the Frobenius norm defined by:

$$\|\mathbf{y}\| = \left[\sum_{i}^{m'} (y_i)^2\right]^{1/2}, \tag{19}$$

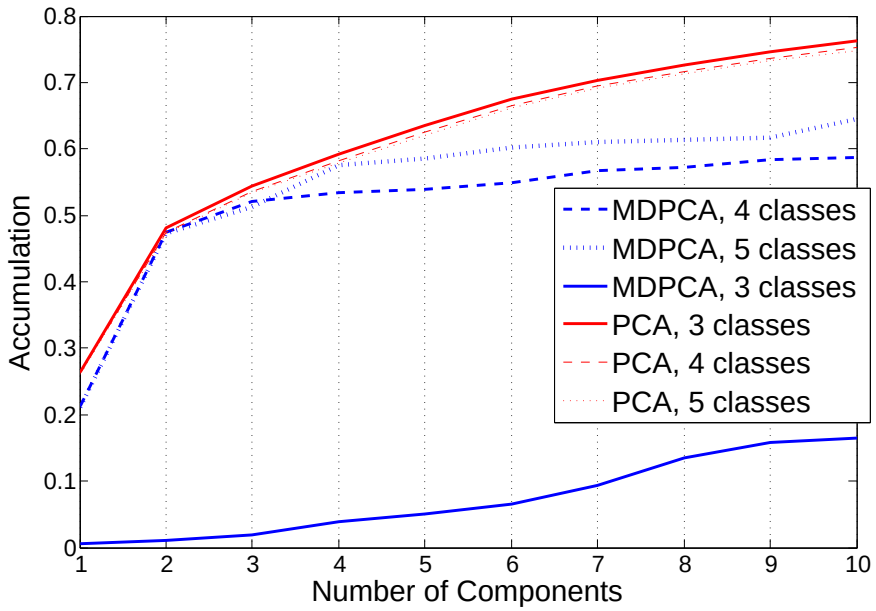
with $\mathbf{y}$ being a m'-dimensional sample.

The Figure 7 shows the RMSE for the reconstruction process for the subspaces given by the focused techniques. It is noticeable that for all experiments, PCA reconstruction performs equal or better than the corresponding Multi-Class DPCA discriminant components for all the simulated values of k. This observation agrees with the accumulated variance reported in Figure 6 which is such that $Variance_{acc}^{PCA,i}(k) \geq Variance_{acc}^{MDPCA,i}(k)$ for all values of k and i.

From Figure 7, for $1 \leq k \leq 2$ the $RMSE^{PCA,4}(k)$ is approximately equal to $RMSE^{MDPCA,4}(k)$. For $k > 2$ we notice that $RMSE^{PCA,4}(k) < RMSE^{MDPCA,4}(k)$.



(a)



(b)

Figure 6: (a) Total variance explained by PCA and Multi-Class DPCA (MDPCA) principal components, computed by expression (17). (b) Analogous, but with a zoom in the first ten components.

An analogous behaviour is observed for the five-class experiments. This observations are in accordance with the corresponding accumulated variances: $Variance_{acc}^{PCA,4}(k) \approx Variance_{acc}^{MDPCA,4}(k)$ for $k < 2$, $Variance_{acc}^{PCA,4}(k) > Variance_{acc}^{MDPCA,4}(k)$ in the range $k > 2$ (analogously for five-class tests). Therefore, while in the classification tasks the PCA method is, in general, outperformed by the Multi-Class DPCA method, in the reconstruction experiments the PCA subspaces become more efficient for almost all the simulated values $2 < k \leq 150$.

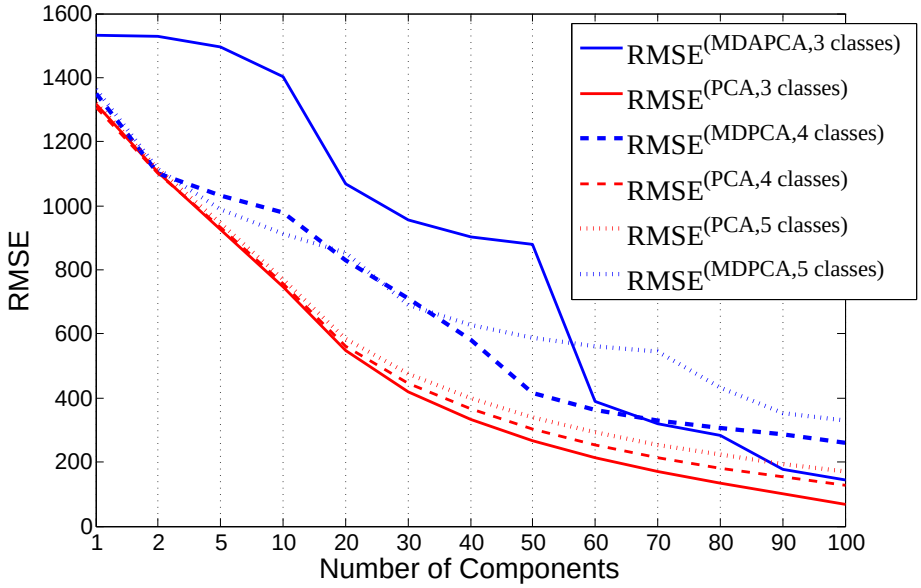


Figure 7: RMSE for PCA and Multi-Class DPCA subspaces.

In order to visualize these facts, we picture in Figure 8 the reconstruction of the happy expression shown in Figure 2.(b) when considering three-class tests and the subspaces generated by the first one ($i = 3$ and $k = 1$ in expression (18)) and by the first 30 principal components ($i = 3$ and $k = 30$ in expression (18)), selected by PCA and Multi-Class DPCA.

The original image is the frontal happy pose shown on Figure 8.(a) and the Figure 8.(b) pictures the average image for the three-class experiment. It is clear that the reconstructions in Figures 8.(c)-(d) are more related with the mean image than with the test sample shown in Figure 8.(a). This observation agrees with the fact that the RMSE shown in Figure 7 is high for small values of k. On the other hand, the reconstructions observed in Figures 8.(e)-(f) are not deviated to the mean. In fact they are much more close to the original image than the results in the middle row. Therefore, we expect smaller values for the RMSE in this case which is in accordance with Figure 7. Besides, Figures 8.(e)-(f) suggest that for $k = 30$ both methods are able to reconstruct the discriminant features. The increasing in the recognition rates for both PCA and Multi-Class DPCA verified in Figure 4, for $i = 3$ and $k = 30$, agrees with this fact.

The noise and artifacts of the images in Figure 8.(c)-(f) make subjective the visual comparison between the corresponding pairs. In order to discard subjective factors, we report in Table 2 the Frobenius norm of the difference $\mathbf{D}_l^k = \mathbf{x}_0 - \mathbf{x}_l^k$ for $k \in \{1, 30\}$ and $l \in \{PCA, MDPCA\}$, computed by expression (19), where $\mathbf{x}_0$ is the original image pictured on Figure 8.(a) and $\mathbf{x}_l^k$ gives the reconstruction results, which corresponds to the images pictured on Figure 8.(c)-(f). The Table 2 shows that, although both Figures 8.(c)-(d) are far from the original image, the

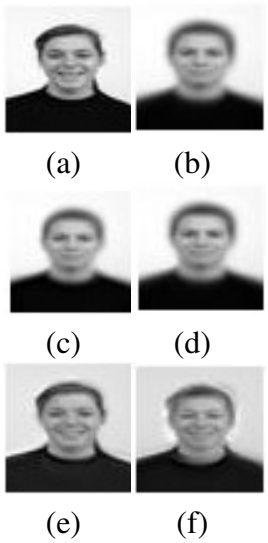


Figure 8: (a) Original image. (b) Average image for the three-class experiment. (c) Reconstruction using the first principal component select by PCA for three-class tests. (d) Analogous result using the first component of Multi-Class DPCA. (e) Reconstruction using the principal PCA subspace of dimension $k = 30$ for three-class experiment. (f) Analogous result for Multi-Class DPCA subspace with dimension $k = 30$.

PCA reconstruction is closer than the Multi-Class DPCA one. On the other hand, both Figures 8.(e)-(f) look like the target but the PCA result seems to have better quality than the Multi-Class DPCA counterpart which is confirmed by the result reported on Table 2.

k	1	30
$\left\|\mathbf{D}_{PCA}^{k}\right\|$	1.0993×10^3	381.9024
$\left\|\mathbf{D}_{MDPCA}^{k}\right\|$	1.2224×10^3	898.9068

Table 2: Reconstruction error, for three-class experiment, computed by expression (19) with $\mathbf{y} = \mathbf{D}_l^k$.

6 CONCLUSIONS AND FUTURE WORKS

This paper proposes an extension of the DPCA technique [Thomaz and Giraldi 2010] for multi-class problems, named Multi-Class DPCA algorithm. The methodology is applied as a ranking method for the PCA components computed using the Radboud facial expression database. Basically, we compute a weakened linear SVM ensemble to get the discriminant weights that are combined through an AdaBoost technique for multilinear discriminant analysis using neutral, happiness, sad, anger and disgust frontal face images.

The facial expressions experiments show that, in general, the principal components selected by Multi-Class DPCA discriminant weights allow higher recognition rates using less linear features than standard PCA. However, the reported experiments show that the PCA performs better for reconstruction than Multi-Class DPCA for most of the selected subspaces. This is a point to be addressed in future works. Also, further work is being undertaken to improve the Multi-Class DPCA in order to avoid the observed performance degradation when increasing the number of classes. Also, comparisons with other state-of-the-art techniques, like LDA [Liera et al. 2014], will be also considered in further works.

REFERENCES

[Burges 1998] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

[Davatzikos 2004] Davatzikos, C. (2004). Why voxel-based morphometric analysis should be used with great caution when characterizing group differences. *NeuroImage*, 23:17–20.

[Devijver and Kittler 1982] Devijver, P. and Kittler, J. (1982). Pattern classification: A statistical approach.

[Filisbino et al. 2013] Filisbino, T., Giraldi, G., and Thomaz, C. (2013). Ranking methods for tensor components analysis and their application to face images. In *Graphics, Patterns and Images (SIBGRAPI), 2013 26th SIBGRAPI - Conference on*, pages 312–319.

[Fodor 2002] Fodor, I. (2002). A survey of dimension reduction techniques. Technical report.

[Fukunaga 1990] Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition*. Academic Press, New York.

[Garcia and Lozano 2007] Garcia, E. and Lozano, F. (2007). Boosting Support Vector Machines. In *Proceedings of International Conference of Machine Learning and Data Mining (MLDM'2007)*, pages 153–167, Leipzig, Germany. IBal publishing.

[Hastie et al. 2001] Hastie, T., Tibshirani, R., and Friedman, J. (2001). The elements of statistical learning. *Springer*.

[Langner et al. 2010] Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D. H. J., Hawk, S. T., and van Knippenberg, A. (2010). Presentation and validation of the radboud faces database. *Cognition & Emotion*, 24(8):1377–1388.

[Li et al. 2008] Li, X., Wang, L., and Sung, E. (2008). Adaboost with svm-based component classifiers. *Eng. Appl. Artif. Intell.*, 21(5):785–795.

[Liera et al. 2014] Liera, A., Gomez, V., and Kappen, H. (2014). Adaptive multiclass classification for brain computer interfaces. *Neural Comput.*, 26:1108–1127.

[Loog et al. 2001] Loog, M., Duin, R. P. W., and Haeb-Umbach, R. (2001). Multiclass linear dimension reduction by weighted pairwise fisher criteria. *IEEE Trans. Patterns Anal. Mach Intell.*, 23(7):762–766.

[Safavi and Chang 2008] Safavi, H. and Chang, C.-I. (2008). Projection pursuit-based dimensionality reduction. *Proc. SPIE*, 6966:69661H–69661H–11.

[Schapire and Singer 1999] Schapire, R. E. and Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. *Mach. Learn.*, 37(3):297–336.

[Swets and Weng 1996] Swets, D. and Weng, J. (1996). Using discriminants eigenfeatures for image retrieval. *IEEE Trans. Patterns Anal. Mach Intell.*, 18(8):831–836.

[Thomaz and Giraldi 2010] Thomaz, C. E. and Giraldi, G. A. (2010). A new ranking method for principal components analysis and its application to face image analysis. *Image Vision Comput.*, 28(6):902–913.

[Turk and Pentland 1991] Turk, M. and Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3:71–86.

[Vapnik 1998] Vapnik, V. N. (1998). *Statistical Learning Theory*. John Wiley & Sons, INC.

[Yildizer et al. 2012] Yildizer, E., Balci, A. M., Hassan, M., and Alhajj, R. (2012). Efficient content-based image retrieval using multiple support vector machines ensemble. *Expert Syst. Appl.*, 39:2385–2396.

[Zhou 2012] Zhou, Z.-H. (2012). *Ensemble Methods: Foundations and Algorithms*. Chapman & Hall/CRC, 1st edition.