Ranking Eigenfaces Through Adaboost and Perceptron Ensembles

Tiene A. Filisbino, Gilson A. Giraldi
Laboratório Nacional de Computação Científica - LNCC
Petrópolis, Brasil
Email: tiene,gilson@lncc.br

Carlos Eduardo Thomaz
Departamento de Engenharia Elétrica
Centro Universitário da FEI
São Bernardo do Campo - Brasil
Email: cet@fei.edu.br

***Abstract*—The fact that principal component analysis (PCA) does not necessarily represent important discriminant directions to separate sample groups motivates the development of the multi-class discriminant principal component analysis (MDPCA). This technique addresses the problem of ranking face features in N-class problems computed by PCA components (eigenfaces). Given a database, the MDPCA builds a linear support vector machine (SVM) ensemble to get the separating hyperplanes that are combined through an AdaBoost technique to determine the discriminant contribution of each PCA feature. In this paper, we follow the MDPCA methodology but we replace the SVM by the linear perceptron as the basic learner in the AdaBoost approach. In the computational experiments we compare the obtained technique, called MDPCA-Perceptron, with the PCA and the original MDPCA through facial expression experiments. Our computational results have shown that the principal components selected by the MDPCA-Perceptron allow competitive recognition rates in lower dimensional spaces with promising results for reconstruction tasks as well.**

***Keywords*-Ranking PCA Components; Separating Hyperplanes; Perceptron; AdaBoost; Face Image Analysis**

I. Introduction

Face image analysis requires 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 [1].

Techniques in linear dimensionality reduction, including the classical principal component analysis (PCA) and the corresponding eigenfaces approach [2], seek for new variables that obey some optimization criterion and can be expressed as linear combination of the original ones [3]. However, linear subspaces selected by dimensionality reduction methods do not necessarily include the most important discriminant directions to separate sample groups [1]. This fact motivates the application and development of other techniques to compute a discriminant subspace.

For the general N-class classification problems, the work [4] presents the multi-class discriminant principal components analysis (MDPCA), that 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 [5], (c) Use AdaBoost techniques [6] to combine the separating SVM hyperplanes in order to determine the global discriminant vector.

The MDPCA 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 [7]. However, the basic learner in the MDPCA procedure is the SVM which can not be considered a weak learner. In fact strong learners have shown performance degradation in boosting algorithms [6]. In order to address this problem, in [8] it is proposed another methodology to build a weakened version of SVM, named WSVM, which discards a percentage μ of the samples in the original data set S to generate the training set. This strategy is applied in the MDPCA algorithm.

In this paper, we keep the MDPCA methodology, but we replace the WSVM technique by the linear perceptron classifier [9]. In this way, in the step (b) we get a a linear perceptron ensemble, based on the "one-against-all" approach. In the step (c) we apply the AdaBoost technique to combine the perceptron hyperplanes to find an accurate global discriminant vector.

It is important to highlight that we do not deal with the problem of computing general discriminant directions that are not principal components. Rather, we apply the idea of using a set of linear classifiers and an ensemble method (AdaBoost, in this case) to compute a discriminate vector that allows to select among the principal components the most discriminant ones.

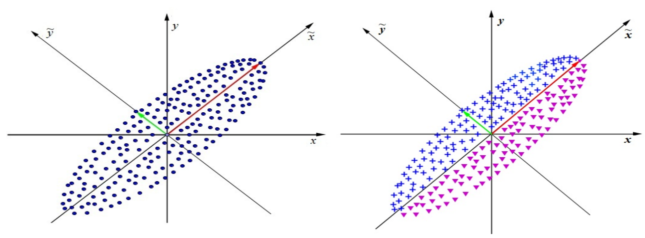
To evaluate the new technique, named MDPCA-Perceptron algorithm, we perform group separation tasks in facial expression experiments involving neutral, happiness, sad, fear, and anger face images. The experiments show that the perceptron can be used as an effective component classifier to generate the discriminant weights for lower subspace dimensions. Furthermore, the reconstruction error of MDPCA-Perceptron is better or equal to the original MDPCA one which is also a promising result.

The paper is organized as follows. Next, in Section II, we review the basic elements behind discriminant analysis. Then, Section III presents the MDPCA-Perceptron approach. The computational experiments are described in Section IV. Finally, in Section V, we conclude the paper, summarizing its main contributions and describing further developments.

II. Discriminant Analysis

It was observed that, since PCA explains the covariance structure of all the data its most expressive components [10], that is, the first principal components with the largest eigenvalues, do not necessarily represent important discriminant directions to separate sample groups. This is particularly true for face image analysis where the PCA components are named eigenfaces [2].

The Figure 1 is a simple example that helps to understand the limitation of PCA to select discriminant features for classification. 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.



(a) (b)

Fig. 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 techniques to compute discriminant subspaces, like MDPCA-Perceptron, which is described next.

III. MDPCA-Perceptron Technique

The MDPCA-Perceptron methodology is summarized by the Algorithm 1. 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 1). The procedure also applies the PCA in its first stages for dimensionality reduction (lines 2-4).

Next, the MDPCA-Perceptron computes a perceptron ensemble, based on the "one-against-all" approach [11]. So, let N be the total number of classes. Each iteration t of the Algorithm 1 constructs one linear perceptron (line 7 of Algorithm 1), in the PCA subspace, using the procedure $Perceptron$, presented in [9], and the training set built in line 6 of Algorithm 1.

The lines (8)-(12) of the Algorithm 1 are based on the AdaBoost philosophy [7] to derive a strong classifier by using the linear combination of perceptron learners $h_1, h_2, \cdots, h_N$:

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

Algorithm 1: MDPCA-Perceptron procedure

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

1 Initialize the homogeneous distribution $D_1(\mathbf{x}_i) = \frac{1}{M}$, $i = 1, 2, \cdots, M$;
2 Determine the PCA projection matrix: P_{pca};
3 Project centered data $\overline{\mathbf{x}}_i = (P_{pca})^T \widetilde{\mathbf{x}}_i \in \mathbb{R}^{m'}$, where $\widetilde{x}_i = x_i - \widehat{x}$ with, $\widehat{x} = \frac{1}{M}\sum_{i=1}^{M} x_i$
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** $t = 1, \ldots$ ***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 $Perceptron(\overline{\Theta}^t, \mathcal{Y}, D_t)$ 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: $\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{m'}$

where α_t does not depend on $\mathbf{x}$ [7].

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 1 which updates the sample distribution D_t (see [7] for details).

The weights $w_{mdpca,i}$ that appear in steps 13-15 of the MDPCA-Perceptron procedure are derived from expression (1). So, following the same idea of [4], the MDPCA-Perceptron determines the discriminant contribution of each feature by investigating the weights $w_{mdpca,i} = \sum_{t=1}^{N} \alpha_t w_{t,i}$. In fact, weights that are estimated to be 0 or approximately 0 have negligible contribution on the discriminant score $H(\mathbf{x})$ given by equation (1), 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, the manner that

AdaBoost combines the weak classifiers gives a straightforward way to compute the discriminant weights (line 13 of the Algorithm 1). In this way, instead of sorting features by selecting the corresponding principal components in decreasing order of eigenvalues, as PCA does, MDPCA-Perceptron selects as the most important features for classification the ones with the highest discriminant weights, that is, $|w_{mdpca,1}| \geq |w_{mdpca,2}| \geq \cdots \geq |w_{mdpca,m'}|$, as performed in lines 14-15 of the algorithm. The output of the MDPCA-Perceptron 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 $|w_{mdpca,i}|$.

IV. Experimental Results

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 [12]. We take five different expressions: the neutral, happiness, sad, anger and fear frontal profile of each person. 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.

Table I lists the 10 principal components with the highest discriminant weights given by the original MDPCA [4], named as MDPCA-SVM in this section, and MDPCA-Perceptron (Algorithm 1), in absolute values, for discriminating the expression samples. From Table I we can observe that the MDPCA-SVM and MDPCA-Perceptron have selected some distant PCA components among the first 10 most discriminant principal ones. We expect some consequences of this fact in the classification experiments, as we will see next.

Expression Experiment: PCA components sorted by MDPCA-SVM and the MDPCA-Perceptron procedure										
MDPCA-SVM	30	27	10	9	6	33	50	14	44	34
MPCA-Perc	30	25	24	27	1	9	8	34	19	7

TABLE I
TOP 10 DISCRIMINANT PRINCIPAL COMPONENTS, RANKED BY MDPCA-SVM AND THE MDPCA-PERCEPTRON.

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:

$$I = \widehat{\mathbf{x}} + \beta \cdot \mathbf{p}_i, \tag{2}$$

where $\widehat{\mathbf{x}}$ is the global mean, $\beta \in \{\pm j \cdot \bar{\lambda}^{0.5}, \quad j = 0, \pm 3\}$, and $\bar{\lambda}$ is the average eigenvalue of the total covariance matrix [1]. 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.

From Table I we notice that the first column do not show difference between the selected principal components. Hence, the Figure 2 illustrates the transformations on the second PCA most expressive component contrasted with the second discriminant principal component selected by MDPCA-Perceptron and MDPCA-SVM to separate facial expressions. We can see in the Figures 2.(a)-(c) that the second PCA most expressive direction captures the changes in gender, which are the major variations of all the training samples.

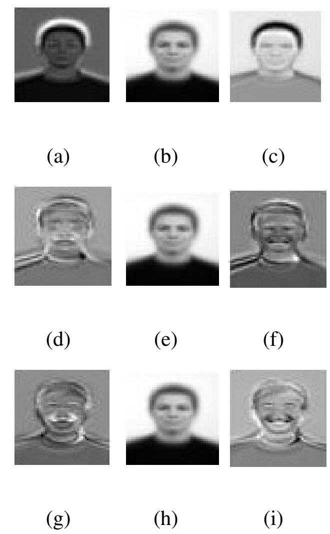


Fig. 2. Visualization of the changes described by the principal directions using expression (2) and Table I: (a)-(c) PCA; (d)-(f) MDPCA-Perceptron; (g)-(i) MDPCA-SVM.

However, when we compare these results with the ones reconstructed by the second discriminant principal component selected by MDPCA-Perceptron (Figures 2.(d)-(f) and by MDPCA-SVM (Figure 2.(g)-(i)), we can see that others PCA components (the 25, 27, according to Table I) carry more information about expression variations then the first PCA ones. We shall analyse the consequences of this fact in the recognition experiments.

The following recognition tasks experiments are carried out using the full rank PCA subspace with all non-zero eigenvalues. We use the 10-fold cross validation method to evaluate the classification performance of PCA, MDPCA-Perceptron and the MDPCA-SVM techniques. In these experiments we have assumed equal prior probabilities and misclassification costs for all the classes. 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:

$$d_i(\mathbf{x}_r) = \sum_{j=1}^{k} \frac{1}{\lambda_j} (x_{rj} - \widehat{x}_{ij})^2, \tag{3}$$

where λ_j is the corresponding covariance matrix 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.

Figure 3 shows the average recognition rates for PCA, MDPCA-Perceptron and the MDPCA-SVM, for five class tasks. We can notice that MDPCA-Perceptron and the

MDPCA-SVM achieve higher recognition rates than the traditional PCA when considering $k < 40$. For $50 \leq k \leq 190$ the recognition rates of PCA subspaces are higher or equal to the MDPCA methodology achieving recognition rate higher than 70% in $k = 120$.

The comparison between MDPCA-Perceptron and MDPCA-SVM shows that they are equivalent for $1 \leq k \leq 5$. Next, MDPCA-SVM performs a bit better than MDPCA-Perceptron in the range $7 < k < 40$. Moreover, in the interval $1 \leq k \leq 50$ the MDPCA-Perceptron obtains the highest recognition rate, among all the other methods, with $k = 40$ achieving recognition rate of 60%. For $55 < k < 180$ MDPCA-Perceptron performs better than MDPCA-SVM. In particular, for $60 < k \leq 96$, MDPCA-Perceptron achieves recognition rates higher than 60%, while MDPCA-SVM performance is below in the same interval.

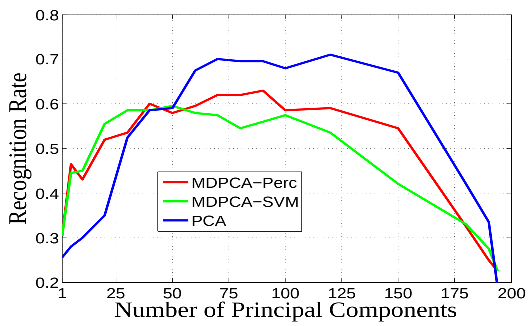


Fig. 3. Average recognition rate of PCA, MDPCA-Perceptron (MDPCA-Perc in the figure), and MDPCA-SVM largest discriminant weights criteria.

Since PCA explains features that most vary in the samples the principal subspaces do not necessarily represent important discriminant directions to separate sample groups [1], [4], which justifies the reported facts about Figure 3. 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:

$$RMSE^{l}(k) = \sqrt{\frac{\sum_{i=1}^{N} ||P.I_k^l.P^T x_i - x_i||^2}{N}}, \quad (4)$$

where $l \in \{PCA, MDPCA - Perceptron, MDPCA - SVM\}$, and I_k^l is a truncated identity matrix that keeps the subspace with dimension k that is selected by PCA, MDPCA-Perceptron, and MDPCA-SVM. In Figure 4, we show the RMSE for the subspaces given by the focused techniques in the five-class experiments. It is noticeable that PCA reconstruction performs better than the MDPCA-Perceptron and MDPCA-SVM discriminant components for all the simulated values of k. Besides, the MDPCA-Perceptron shows equal or lower RMSE values than MDPCA-SVM everywhere but largest than PCA.

V. Conclusion

In this paper we propose the MDPCA-Perceptron methodology for ranking PCA components. We compute the discriminant weights for multi-class discriminant analysis using neutral, happiness, sad, anger and fear frontal face images of the Radboud database. The facial expressions experiments show that the principal components

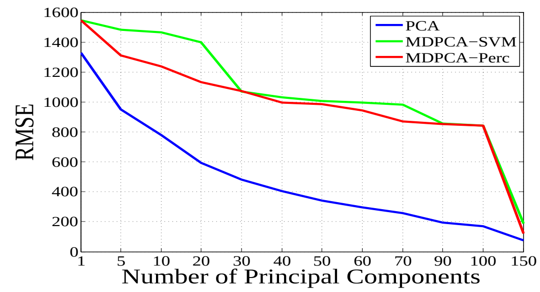


Fig. 4. RMSE for PCA, MDPCA-Perceptron (MDPCA-Perc in the figure), and MDPCA-SVM subspaces.

selected by MDPCA-Perceptron discriminant weights gives competitive recognition rates if compared with the PCA and MDPCA-SVM ones for subspace dimensions in the range $1 \leq k \leq 40$. In terms of reconstruction, Figure 4 shows that the PCA is the best technique followed by the MDPCA-Perceptron. Therefore, in applications that requires suitable (not the best) classification and good reconstruction the MDPCA-Perceptron using fill features can fulfill both requirements.

Further work is being undertaken to test the methodology using AdaBoost.M2 technique [13] as a direction to improve the classification performance, as well as, to use other techniques to compute discriminant vectors, as LDA and Partial Least Squares (PLS)[1].

References

[1] R. Brereton, *Chemometrics for pattern recognition*. John Wiley & Sons, 2009.

[2] M. A. Turk and A. P. Pentland, “Face recognition using eigenfaces,” in *IEEE Conf. on Comp. Vis. and Patt. Recog.*, Jun 1991, pp. 586–591.

[3] J. P. Cunningham and Z. Ghahramani, “Linear dimensionality reduction: Survey, insights, and generalizations,” *J. of Mach. Learn. Research*, vol. 16, pp. 2859–2900, 2015.

[4] T. Filisbino, D. Leite, G. Giraldi, and C. Thomaz, “Multi-class discriminant analysis based on svm ensembles for ranking principal components,” in *36th Ibero-Latin Am. Cong. on Comp. Meth. in Eng. (CILAMCE)*, Nov 2015.

[5] C. J. C. Burges, “A tutorial on support vector machines for pattern recognition,” *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[6] X. Li, L. Wang, and E. Sung, “Adaboost with svm-based component classifiers,” *Eng. Appl. Artif. Intell.*, vol. 21, no. 5, pp. 785–795, Aug. 2008.

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

[8] E. Garcia and F. Lozano, “Boosting Support Vector Machines,” in *Proceedings of International Conference of Machine Learning and Data Mining (MLDM’2007)*. Leipzig, Germany: IBal publishing, Jul. 2007, pp. 153–167.

[9] R. Beale and T. Jackson, *Neural Computing*. MIT Press, 1994.

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

[11] E. Yildizer, A. M. Balci, M. Hassan, and R. Alhajj, “Efficient content-based image retrieval using multiple support vector machines ensemble,” *Expert Syst. Appl.*, vol. 39, pp. 2385–2396, Feb. 2012.

[12] O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S. T. Hawk, and A. van Knippenberg, “Presentation and validation of the radboud faces database,” *Cognition & Emotion*, vol. 24, no. 8, pp. 1377–1388, 2010.

[13] Y. Freund and R. E. Schapire, “A decision-theoretic generalization of on-line learning and an application to boosting,” *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.