Face Detection Using an SVM Trained in Eigenfaces Space

Vlad Popovici[1] and Jean-Philippe Thiran[1]

Signal Processing Institute
Swiss Federal Institute of Technology Lausanne
CH-1015 Lausanne, Switzerland
{Vlad.Popovici, JP.Thiran}@epfl.ch
http://ltswww.epfl.ch

Abstract. [1] The central problem in the case of face detectors is to build a face class model. We present a method for face class modeling in the eigenfaces space using a large-margin classifier like SVM. Two main issues are addressed: what is the required number of eigenfaces to achieve a good classification rate and how to train the SVM for a good generalization. As the experimental evidence show, generally one needs less eigenfaces than usually considered. We will present different strategies for choosing the dimensionality of the PCA space and discuss their effectiveness in the case of face-class modeling.

1 Introduction

Human face detection is usually the first task performed in a face recognition system. Its performances significantly influence the overall quality of the system. In spite of considerable attention that it has received, the problem of reliable face detection remains open. The difficulty stems from the fact that face detection is a problem of categorization: the system must recognize objects belonging to a large class, not just previously seen entities. While, theoretically, the set of all human faces is finite, practically it is impossible to have access to all its instances. However, as all the faces share the same structure, there must be an underlying model that generates all instances of the face class. The problem is then to find (an approximation of) this model and a good classification function.

One of the most effective approaches is to model the set of available faces as a sequence of linear approximations. The best (in the sense of least squares) such approximation is given by Principal Component Analysis (PCA) [1]. The use of PCA in the context of face modeling dates for more than a decade ago ([2],[3]) and proved its capabilities in different contexts like face detection or face recognition. However, in the case of most applications a simple decision rule, e.g. a simple threshold (like in the case of distance–from–feature–space – see below), or a linear classifier (LDA) [4] is used to discriminate between faces and non-faces or for face recognition. Another problem is how to choose the number of required principal components. While in the context of

[1] Work partially performed in the BANCA project of the IST European program with the financial support of the Swiss OFES and with the support of the IM2–NCCR of the Swiss NFS.

face recognition it makes much sense to use the reconstruction error as an indication of the number of components, in the case of face-class modeling this is not so evident. In fact, as it will be shown below, one needs less principal components to achieve a good performance than usually is considered.

This paper tries to address both problems of selecting the number of components and designing a more flexible discriminant function. Its structure is as follows: the first two sections address the theoretical aspects of the classifier used (SVM) and of the eigenfaces space while the third section is dedicated to the experimental results. Finally, we draw some conclusions in the last section.

2 Eigenfaces for face modeling

2.1 Principal Component Analysis (PCA)

Let $\mathbf{x}_1, \ldots, \mathbf{x}_l \in \mathbb{R}^n$ be a set of $n-$dimensional vectors and consider the following linear model for representing them

$$\mathbf{x} = W_{(k)}\mathbf{z} + \mu \tag{1}$$

where $W_{(k)}$ is a $n \times k$ matrix, $\mathbf{z} \in \mathbb{R}^k$ and $\mu \in \mathbb{R}^n$. For a given $k < n$, the PCA can be defined ([1]) as the transformation $W_{(k)}$ whose column vectors $\mathbf{w}_j$, called *principal axes*, are those orthonormal axes onto which the retained variance under projection is maximal. It can be shown that the vectors $\mathbf{w}_j$ are given by the dominant k eigenvectors of the sample covariance matrix[2] $S = \frac{1}{l}\sum_l(\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)'$ such that $S\mathbf{w}_j = \lambda \mathbf{w}_j$ and where μ is the sample mean. The vector $\mathbf{z}_i = W'_{(k)}(\mathbf{x}_i - \mu)$ is the $k-$dimensional representation of the observed vector $\mathbf{x}_i$.

The projection defined by PCA is optimal in the sense that amongst the $k-$dimensional subspaces, the one defined by the columns of $W_{(k)}$ minimizes the reconstruction error $\sum_i \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2$ where $\hat{\mathbf{x}}_i = W_{(k)}\mathbf{z}_i + \mu$.

2.2 Probabilistic PCA (PPCA)

The PPCA ([5]) also assumes a linear model for the observed data

$$\mathbf{x} = W_{(k)}\mathbf{z} + \mu + \epsilon \tag{2}$$

(compare it with (1)) which is closely related to the factor analysis model, but it differs from it in the assumptions made about the density functions generating $\mathbf{z}$ and ϵ:

$$p(\mathbf{z}) \sim \mathcal{N}(0, \sigma^2 I) \tag{3}$$

$$p(\epsilon) \sim \mathcal{N}(0, I) \tag{4}$$

Under this model, the probability of observing the vector $\mathbf{x}$ is

$$p(\mathbf{x}|W, \mu, \sigma^2) \sim \mathcal{N}(\mu, WW' + \sigma^2 I) \tag{5}$$

[2] We denote with a prime symbol the transpose of a matrix or a vector.

For this model, an elegant EM algorithm for estimating the parameters of the model is given in [5]. A similar model was also discussed in [6] in the context of object detection.

Here we are interested in the approach taken in [7] for estimating the underlying dimensionality. Starting from the above model, it can be shown [7] that

$$p\left(\{\mathbf{x}_i\}_{i=1}^{l} \,|k\right) \approx \left(\prod_{j=1}^{k} \lambda_j\right)^{-\frac{l}{2}} (\hat{\sigma}^2)^{-\frac{l(n-k)}{2}} l^{-\frac{m+k}{2}} \tag{6}$$

where $m = \frac{n(n-1)}{2} - \frac{(n-k)(n-k-1)}{2}$ and λ_j are the eigenvalues of the sample covariance matrix. (6) is the Bayesian Information Criterion (BIC) approximation of the likelihood (5). In one set of experiments we will use this criterion for choosing the PCA dimensionality.

2.3 Eigenfaces

Let I denote a $n_1 \times n_2$ gray-scale image. By considering its pixels in lexicographic order, we build a vector x of size $n = n_1 n_2$, which can be seen as a point in $\mathbb{R}^n$. When performing PCA we will use these vectors instead of the original set of images. In the context of face detection and/or recognition, the eigenvectors of the sample covariance matrix are called *eigenfaces* ([8]). A large number of methods rely on the *distance from feature space (DFFS)*

$$\mathrm{dffs}(\mathbf{x}) = ||\mathbf{x} - \hat{\mathbf{x}}||^2 = ||\mathbf{x} - \mu||^2 - ||\mathbf{z}||^2 \tag{7}$$

and/or *distance in feature space (DIFS)*

$$\mathrm{difs}(x) = ||\mathbf{z}||^2 \tag{8}$$

for estimating the class membership. For example, in [6] the following distance is derived under the assumptions (2,3,4):

$$d(\mathbf{x}) = \sum_{i=1}^{k} \frac{z_i^2}{\lambda_i} + \frac{n-k}{\sum_{i=k+1}^{n} \lambda_i} \mathrm{dffs}(\mathbf{x}) \tag{9}$$

3 An Overview of Support Vector Machines

In this section we briefly sketch the SVM algorithm and its motivation. A more detailed description of SVM can be found in [9], [10].

Let us consider first the simple case of linearly separable data. We are searching an *optimal separating (hyper–)plane* [3]

$$\langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \tag{10}$$

[3] We use $\langle \cdot, \cdot \rangle$ to denote the inner product operator

which minimizes the VC confidence term while providing the best generalization. The decision function is

$$f(\mathbf{x}) = \mathrm{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right) \tag{11}$$

Geometrically, the problem to be solved is to find the hyperplane that maximizes the sum of distances to the closest positive and negative training examples. The distance is called *margin* and the optimal plane is obtained by maximizing $\frac{2}{\|\mathbf{w}\|}$ or, equivalently, by minimizing $\|\mathbf{w}\|^2$ subject to $y_i(\langle \mathbf{w}, \mathbf{x} \rangle + b) \geq 1$. Suppose now that the two classes overlap in feature space. One way to find the optimal plane is to relax the above constraints by introducing the *slack variables* ξ_i and solving the following problem (using 2-norm for the slack variables):

$$\min_{\xi, \mathbf{w}, b} \|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \xi_i^2 \tag{12}$$

$$\text{subject to} \qquad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \quad \forall i = 1, \ldots, l \tag{13}$$

where C controls the weight of the classification errors ($C = \infty$ in the separable case).

By introducing the Lagrange multipliers, we obtain the primal and the dual Lagrangian forms

$$L_P = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \xi_i^2 - \sum_{i=1}^{l} \alpha_i \left[y_i \left(\langle \mathbf{w}, \mathbf{x}_i \rangle \right) - 1 - \xi_i \right] \tag{14}$$

$$L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \frac{1}{C} \sum_{i=1}^{l} \alpha_i. \tag{15}$$

where $\alpha_i \geq 0$. The solution of the primal problem is linked to the solution of the dual by $\mathbf{w} = \sum_i y_i \alpha_i \mathbf{x}_i$.

We can express now the decision function as a function of α:

$$f(\mathbf{x}) = \mathrm{sgn}\left(\sum_{i \in S} y_i \alpha_i \langle \mathbf{x}, \mathbf{x}_i \rangle + b \right) \tag{16}$$

where $S = \{i \mid \alpha_i > 0\}$. The vectors $\mathbf{x}_i, i \in S$ are called *support vectors* and are the only examples from the training set that affect the shape of the separating boundary.

In practice however, a linear separating plane is seldom sufficient. To generalize the linear case one can project the input space into a higher–dimensional space in the hope of a better training–class separation. In the case of SVM this is achieved by using the so–called ”kernel trick”. In essence, it replaces the inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ in (15) and (16) with a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$. As the data vectors are involved only in this inner products, the optimization process can be carried out in the feature space directly. Some of the most used kernel functions are:

$$\text{the polynomial kernel} \qquad K(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^d \tag{17}$$

$$\text{the RBF kernel} \qquad K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2) \tag{18}$$

4 Proposed method and Experiments

Relying on eigenfaces for describing the face model is an appealing technique. Not only we reduce the dimensionality of the input space, thus needing less examples for training the classifiers, but also the eigenfaces proved to be robust features in real-world applications.

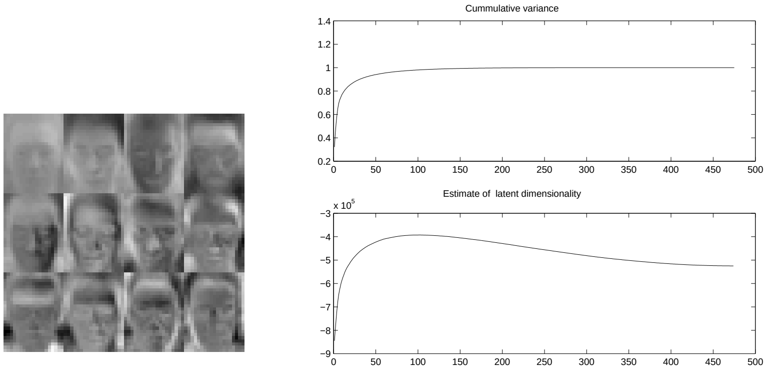
We want to benefit from those advantages while going beyond the DFFS-like classification methods. To this end, we propose to use a SVM to directly model the face class boundary. There are a number of issues that must be addressed like how many eigenfaces are needed for a good face class model and what kernel should be employed for SVM. We will analyze different alternatives of choosing the PCA dimensionality and discuss the performances of the SVM for each of those choices.

4.1 Experiments

In the following we will discuss a set of experiments that were performed to study the performance of SVM-based classifiers in the eigenfaces space. As pointed out before, the main problem in the case of face detection is finding a good model for the entire class of faces. As such, we concentrated mainly on the face/non–face classification task.

The face dataset used was a subset of BANCA database ([11]), consisting of 6540 images. Faces were cropped out from the images and rescaled to 19×25. The positive example set (faces) consisted of 3 subsets, labelled *g1*, *g2* and *wm*, respectively, containing different individuals. The *g1* and *g2* sets contained 3120 images each, for 26 individuals per set, recorded in 12 sessions (10 images per session) covering 3 different environments (different illumination conditions, different pose and background). The set *wm* contained 300 images of 30 different individuals recorded in 3 sessions (10 individuals per session). For training, the positive example set consisted always of 684 images from *wm* and either *g1* or *g2* (2 images per individual per session, 26 individuals from either *g1* or *g2* and 30 individuals from *wm*); for testing we used all the images of individuals not present in training set (i.e. if we used images from *g1* in training then all *g2* set was used for testing and the reciprocal), so we had 3120 face images for tests. The negative examples were collected from various images not containing human faces by bootstrapping some initial versions of the classifiers. In all, there were 19500 non-face images, splited in two sets of 7000 and 12500 for training and testing respectively. The classification results that will be presented are the average classification rates obtained. Figure 1 presents the first eigenfaces from the set of principal axes obtained by performing PCA on the positive training set and the estimation of the latent dimensionality of the eigenface space.

First we studied the influence of the PCA dimensionality on the performance of the classifier. We trained a SVM with a RBF kernel (see (18)), keeping its parameters (i.e. γ and C) constant and we varied the number of eigenfaces used to construct the ”face space”. Figure 2 shows the variability of different performance indices. As one would expect, while the training performances keep increasing, the testing results show a peak in true positive rate. This peak coincides with the estimated latent dimensionality (102). However, using so many eigenfaces impacts on the speed of the computations. In real applications one has to trade off some performance points for a speedup of the



(a) First 12 eigenfaces.

(b) Cummulative variance and latent dimensionality

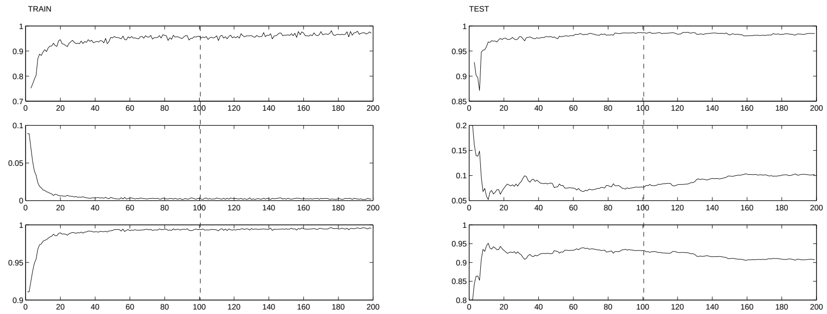
Fig. 1. Eigenfaces and cummulative variance. 1(a) The first 12 eigenfaces correspond roughly to 80% of total variantion of face set. 1(b) Cummulative variance and latent dimensionality estimation by BIC approximation (Eq.6)

detection. For a faster detection, it seems reasonable to choose only 20 eigenfaces and then to tune the classifier in this reduced space.

We will further investigate the classification performances by tuning the classifiers for 3 different dimensionalities: 20, 36 (which corresponds roughly to 90% of total variation) and 102 (as suggested by BIC) eigenfaces. We trained two different SVM, one with a polynomial kernel and another one with a RBF kernel (equations (17,18)), varying their parameters. The results are presented in figure 3. As can be seen, adding more eigenfaces in the representation improves up to a point the results. However, having too many eigenfaces leads to less stable behavior of the SVM (in the case of the polynomial kernel) or even degrades the performances. This is due to both the over–fitting effect that may appear in training and to the limited number of training samples used. Interestingly, even the difference between the two cases (20 and 36 eigenfaces respectively) is not so important if we consider that in the first case we have almost half of the number of eigenfaces (which corresponds to approximately 85% of total variation).

The best classification rates are summarized in Table 1. For comparison, the classification rates obtained with a simple threshold based classifier (using the distance (9) are given in the last row, even if the difference in complexity between the two classifiers makes the comparison unfair.

Finally we used a classifier trained as above, using only 20 eigenfaces and a RBF kernel, for a large scale test on real-world data (the entire set of images from the English part of BANCA database). We scanned the images with a sliding window at different scales. The overall performace (aggregated over the 3 conditions) was about 95%. More detailed results will be presented in the final version of the paper.



(a) Training results

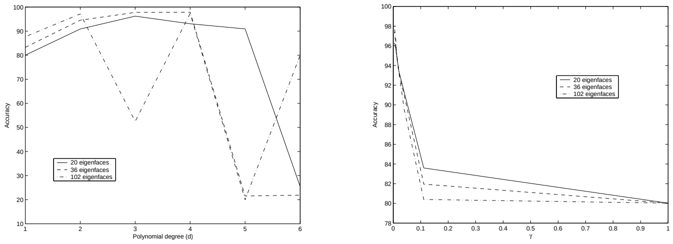
(b) Testing results

Fig. 2. Training and testing result using a RBF kernel. The dashed line indicates the estimated dimensionality of the PCA space. The pannels show three performance factors (from top to bottom): true positive rate, false positive rate and overall accuracy with respect to the number of selected eigenfaces.

5 Conclusions

In this paper we presented a method for face class modeling in eigenfaces space. The method relies on a SVM for class boundary modeling, being able to implement highly nonlinear (in eigenfaces space) decision functions.

Another issue that we have addressed was the problem of the number of eigenfaces needed to achieve good performances. We have compared different approaches like the ”90%” rule-of-thumb or the more principled BIC approximation. As the experiments have shown, generally one needs less eigenfaces than suggested by those rules to reach an acceptable level of accuracy. Beyond that, one needs a large number of additional



(a) Polynomial SVM

(b) RBF SVM

Fig. 3. Accuracy of two SVM on the test set. The horizontal axis represents the values of the kernel parameter.

Classifier	Number of eigenfaces		
	20	36	102
Polynomial SVM	96.21%	97.86%	97.35%
RBF SVM	96.30%	97.41%	97.93%
Distance-based	75.91%	77.38%	78.85%

Table 1. Top performances on the test set. Kernel parameters were $d = 3, d = 4, d = 4$ for the polynomial kernel and $\gamma = 0.015, \gamma = 0.017, \gamma = 0.015$ for the RBF kernel, respectively.

eigenfaces for a significant improvement. An interesting outcome is the coincidence of the number of eigenfaces needed for the highest true positive rate with the latent dimensionality suggested by BIC. However, this criterion produces a largely overestimate number of eigenfaces if we take into account the overall accuracy of the classifier.

References

1. Hotteling, H.: Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology (1933) 417–441, 498–520
2. Sirovich, L., Kirby, M.: Low-dimensional procedure for the characterization of human faces. Journal of the Optical Society of America A **4** (1987) 519–524
3. Turk, M.A., Pentland, A.P.: Face recognition using eigenfaces. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition. (1991) 586–591
4. Zhao, W., Chellappa, R., Krishaswamy, A.: Discriminant analysis of principal components for face recognition. In: Proceedings of the 3rd International Conference on Automatic Face and Gesture Recognition. (1998)
5. Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. Technical Report NCRG/97/010, Neural Computing Research Group, Dept. of Computer Science and Applied Mathematics, Aston University (1997)
6. Moghaddam, B., Pentland, A.: Probabilistic visual learning for object detection. In: Proc. of the 5th International Conference on Computer Vision. (1995) 786–793
7. Minka, T.P.: Automatic choice of dimensionality for PCA. Technical Report 514, M.I.T. Media Laboratory Perceptual Computing Section (2000)
8. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience **3** (1991) 71–86
9. Vapnik, V.: The Nature of Statistical Learning Theory. Springer Verlag (1995)
10. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press (2000)
11. Bengio, S., Bimbot, F., Mariéthoz, J., Popovici, V., Porée, F., Bailly-Baillière, E., Matas, G., Ruiz, B.: Experimental protocol on the BANCA database. IDIAP-RR 05, IDIAP (2002)
12. Sung, K., Poggio, T.: Example-based learning for view-based human face detection. IEEE Transaction on Pattern Analysis and Machine Intelligence **20** (1998) 39–51
13. Osuna, E., Freund, R., Girosi, F.: Training support vector machines: an application to face detection. In: Proceedings of Intl. Conference on Computer Vision and Pattern Recognition (CVPR). (1997)
14. Penev, P.S., Sirovich, L.: The global dimensionality of face space. In: Proceedings of the 4th Intl. Conference on Automatic Face and Gesture Recognition, IEEE CS (2000) 264–270
15. Roth, V., Steinhage, V.: Nonlinear discriminant analysis using kernel functions. In Solla, S., Leen, T., Müller, K.R., eds.: Advances in Neural Information Processing Systems. Volume 12., MIT Press (1999) 568–574