Face Class Modeling in Eigenfaces Space

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

[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. We [2] present a method for face class modeling in the eigenfaces space using a large-margin classifier like SVM. Another issue addressed is how to select the number of eigenfaces to achieve a good classification rate. As the experimental evidence show, generally one needs less eigenfaces than usually considered. We will present different strategies 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. 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. Recently, numerous intensity-based methods have been proposed to detect human faces in a single image or a sequence of images. A few significant approaches are briefly reviewed below.

One of the most representative approaches for the class of neural networks–based face detectors is the work reported by Rowley et. al. in [1]. Their system has two major components: a face detector and a final decision module. The face detector uses multiple neural networks to detect the 20×20 candidate regions in an image that is scanned at different positions and scales. The second component is used to merge the overlapping regions and arbitrate between the outputs of multiple networks. Later, a new module (a router) has been added, which had the task of detecting the orientation of the possible face and rotate it in canonical orientation.

Colmenarez and Huang used the Kullback relative information for maximizing the discrimination between positive and negative examples of faces [2]. The two classes were characterized in terms of their probability distribution functions, approximated by a family of discrete Markov processes. The learning process wass converted into an

[2] 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.

optimization problem to select the Markov process that maximized the information–based discrimination between the two classes.

Sung and Poggio have developed a clustering and distribution-based system for face detection [3]. There are two main components in their system: a model of the face/non–face patterns distribution and a decision making module. The two class distributions are each approximated by six Gaussian clusters. For a given pattern two types of distances are computed: first a normalized Mahalanobis distance between the test pattern and the 12 cluster centroids, measured in the subspace spawned by the first 75 eigenvectors of each cluster. The second distance is a Euclidean distance between the test pattern and its projection onto the 75–dimensional subspaces. Finally, a multilayer perceptron is trained to dicriminate the patters using the 12 pairs of distances.

A naive Bayes classifier based on local appearance and position of the face pattern at different resolutions is described by Schneiderman and Kanade in [4]. The face samples are decomposed in four rectangular subregions which are then projected to a lower dimensional space using PCA and quantized into a finite set of patterns. The statistics of local appearances are estimated independently from the samples. By discarding the statistical dependencies between the regions, they obtained better estimates of the density functions and a functional form for the posterior probability.

Osuna et. al. developed a face detector based on SVM that worked directly on the intensity patterns [5]. A brief description of the SVM is given in this paper also. The large scale tests they performed showed a slightly lower error rate than the system of Sung and Poggio, while running approximately 30 times faster.

In the following we will address the problem of face class modeling in the eigenfaces space. We will present a method that avoids estimating the class conditional probabilities by directly focusing on modelin the face class boundary, by means of Support Vector Machines. Another issue to be addressed relates to estimating the necessary number of eigenfaces for a good classificatio performance. Also, we will discuss a technique of postprocessing the outputs of SVM in order to be able to interpret them as *a posterior* probability approximations. The paper is structured 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 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 [6], [7].

Let $\{(\mathbf{x}_i, y_i)|i = 1, \ldots, l\} \subset \mathbb{R}^n \times \{-1, +1\}$ be a set of examples. From a practical point of view, the problem to be solved is to find that hyperplane that correctly separates the data while maximizing the sum of distances to the closest positive and negative points (i.e. *the margin*). The hyperplane is given by[3]:

$$h_{\mathbf{w},b}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b = 0 \tag{1}$$

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

and the decision function is

$$f(\mathbf{x}) = \operatorname{sgn}(h_{\mathbf{w},b}(\mathbf{x})) = \operatorname{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right) \tag{2}$$

In the case of linearly separable data, maximizing the margins means to maximize $\frac{2}{\|\mathbf{w}\|}$ or, equivalently, to minimize $\|\mathbf{w}\|^2$, 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):

$$\min_{\xi, \mathbf{w}, b} \|\mathbf{w}\|^2 + C \sum_{i=1}^{l} \xi_i^2 \tag{3}$$

$$\text{subject to} \qquad y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \quad \forall i = 1, \ldots, l \tag{4}$$

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

This problem is solved by means of Lagrange multipliers method. Let $\alpha_i \geq 0$ be the Lagrange multipliers solving the problem above, then the separating hyperplane, as a function of α_i, is given by

$$h_{\alpha_i, b}(\mathbf{x}) = \sum_{i, \alpha_i > 0} y_i \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle + b \tag{5}$$

Note that usually only a small proportion of α_i are non-zero. The training vectors $\mathbf{x}_i$ corresponding to $\alpha_i > 0$ are called *support vectors* and are the only training vectors influencing 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". Basically, it replaces the inner product $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ 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:

$$\text{the polynomial kernel} \qquad K(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + 1)^d \tag{6}$$

$$\text{the RBF kernel} \qquad K(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2) \tag{7}$$

3 Eigenfaces for face modeling

3.1 Principal Component Analysis (PCA) and Eigenfaces

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

$$\mathbf{x} = W_{(k)} \mathbf{z} + \mu \tag{8}$$

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 ([8]) 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[4] $S = \frac{1}{l}\sum_l (\mathbf{x}_i - \mu)(\mathbf{x}_i - \mu)'$ 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 $\hat{\mathbf{x}}_i = W_{(k)}\mathbf{z}_i + \mu$.

Now let us view an image as a vector in $\mathbb{R}^n$ space by considering its pixels in lexicographic order. Then the PCA method can be applied to images as well, and in the case of face images the principal directions are called *eigenfaces* [9],[10]. Traditionally, the distance between a given image and the class of faces has been decomposed in two orthogonal components: the *distance in feature space* (corresponding to the projection onto the lower dimensional space) and the *distance from feature space (DFFS)* (accounting for the reconstruction error).

3.2 Probabilistic PCA (PPCA) and Latent Dimensionality Estimation

The PPCA ([11]) also assumes a linear model for the observed data

$$\mathbf{x} = W_{(k)}\mathbf{z} + \mu + \epsilon \tag{9}$$

(compare it with (8)) 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:

$$p(\mathbf{z}) \sim \mathcal{N}(0, \sigma^2 I) \tag{10}$$

$$p(\epsilon) \sim \mathcal{N}(0, I) \tag{11}$$

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

$$p(\mathbf{x}|W, \mu, \sigma^2) \sim \mathcal{N}(\mu, WW' + \sigma^2 I) \tag{12}$$

For this model, an elegant EM algorithm for estimating the parameters of the model is given in [11]. A similar model was also discussed in [12] in the context of object detection.

Here we are interested in the approach taken in [13] for estimating the underlying dimensionality. Starting from the above model, it can be shown [13] that

$$p\left(\{\mathbf{x}_i\}_{i=1}^l | k\right) \approx \left(\prod_{j=1}^k \lambda_j\right)^{-\frac{l}{2}} (\hat{\sigma}^2)^{-\frac{l(n-k)}{2}} l^{-\frac{m+k}{2}} \tag{13}$$

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. (13) is the Bayesian Information Criterion (BIC) approximation of the likelihood (12). In one set of experiments we will use this criterion for choosing the PCA dimensionality.

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

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 more robust features in real-world applications than the raw pixel values.

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.

One disadvantage of using SVM for classification stems from the fact that its output is either $+1$ or -1 without any confidence measure. It is well known the problem of multiple detections that is usually solved by an arbitration technique. We would like to have a means of interpreting the SVM outputs in a probabilistic manner, i.e. we would like to calibrate the outputs to model the posterior probabilities. There are a number of approaches proposed, and we decided to fit a sigmoid function to the values of $h(\mathbf{x})$. This means we have to find those values for α and β from

$$p(\mathbf{x}) = \frac{1}{1 + \exp(\alpha h(\mathbf{x}) + \beta)} \tag{14}$$

that minimize that the log-likelihood of the training data (cross-entropy error function):

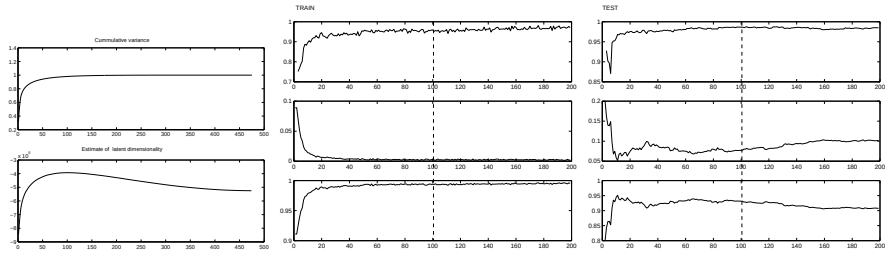
$$-\sum_i (\tilde{y}_i \log(p(\mathbf{x}_i)) + (1 - \tilde{y}_i) \log(1 - p(\mathbf{x}_i))) \tag{15}$$

where the new labels are $\tilde{y}_i = 0.5(1 + y_i)$. The new classification rule will be to assign $\mathbf{x}$ to class "+1" if $p(\mathbf{x}) \geq 0.5$ and to class "-1" otherwise. Also, we will use the values $p(\mathbf{x})$ to arbitrate between the multiple detections.

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 [14], containing 6540 images. Faces were cropped out from the images and rescaled to 19×25. The training set consisted of 684 images of faces and 7000 of non-faces, while the testing set consisted of 3120 images of faces and 12500 of non-faces. The identities of faces in the testing set were different from those in the training set. For a detailed description of the database structure and contents, the reader is referred to [14]. 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.



(a) Cummulative variance and latent dimensionality

(b) Training results

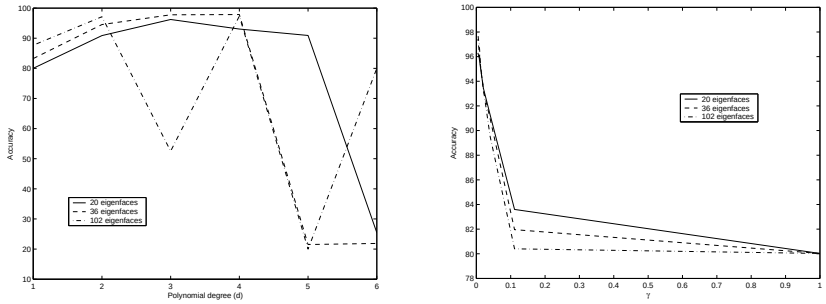
(c) Testing results

Fig. 1. 1(a) Cummulative variance and latent dimensionality estimation by BIC approximation (Eq.13). 1(b)– 1(c)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.

First we studied the influence of the PCA dimensionality on the performance of the classifier. We trained a SVM with a RBF kernel (see (7)), keeping its parameters (i.e. γ and C) constant and we varied the number of eigenfaces used to construct the "face space". Figure 1(b)–1(c) 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 detection. For a faster detection, it seems reasonable to choose only 20 eigenfaces and then 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 (6,7)), varying their parameters. The results are presented in figure 2. 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. Also, for the trained classifiers we performed a set of tests in which we postprocessed the outputs by fitting the sigmoid function, as described above. For comparison, the classification rates obtained



(a) Polynomial SVM (b) RBF SVM

Fig. 2. Accuracy of two SVM on the test set. The horizontal axis represents the values of the kernel parameter.

with a simple threshold based classifier (using the distance from feature space) are given in the last row, even if the difference in complexity between the two classifiers makes the comparison unfair. It is interesting to notice that fitting the sigmoid on the outputs

Classifier	Number of eigenfaces					
	Without postprocessing			With postprocessing		
	20	36	102	20	36	102
Polynomial SVM	96.21%	97.86%	97.35%	96.67%	98.43%	97.30%
RBF SVM	96.30%	97.41%	97.93%	96.81%	97.93%	98.38%
Distance-based	75.91%	77.38%	78.85%	N/A	N/A	N/A

Table 1. Top performances on the test set

of the SVM slightly decreased the error rate in all but one case. However, using this technique we have also the posibility to interpret the reliability of the outputs.

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 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. Finally, we discuss a postprocessing technique for for the SVM outputs which is intended to be used for arbitrating between multiple possible detections. Different alternative for modeling the SVM margin are currently under investigation.

References

1. Rowley, H.A., Baluja, S., Kanade, T.: Human face detection in visual scenes. In Touretzky, D.S., Mozer, M.C., Hasselmo, M.E., eds.: Advances in Neural Information Processing Systems. Volume 8., The MIT Press (1996) 875–881
2. Colmenarez, A., Huang, T.: Face detection with informationbased maximum discrimination. In: Proceedings of Computer Vision and Pattern Recognition. (1997) 782–787
3. 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
4. Schneiderman, H., Kanade, T.: Probabilistic modeling of local appearance and spatial relationship for object recognition. In: Proceedings of Computer Vision and Pattern Recognition. (1998) 45–51
5. Osuna, E., Freund, R., Girosi, F.: Training support vector machines: an application to face detection. In: Proceedings of Computer Vision and Pattern Recognition. (1997)
6. Vapnik, V.: The Nature of Statistical Learning Theory. Springer Verlag (1995)
7. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press (2000)
8. Hotteling, H.: Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology (1933) 417–441, 498–520
9. 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
10. Turk, M., Pentland, A.: Eigenfaces for recognition. Journal of Cognitive Neuroscience **3** (1991) 71–86
11. 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)
12. Moghaddam, B., Pentland, A.: Probabilistic visual learning for object detection. In: Proc. of the 5th International Conference on Computer Vision. (1995) 786–793
13. Minka, T.P.: Automatic choice of dimensionality for PCA. Technical Report 514, M.I.T. Media Laboratory Perceptual Computing Section (2000)
14. 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)
15. Yang, M., Kriegman, D., Ahuja, N.: Face detection using multimodal density models. Computer Vision and Image Understanding (2001) 264–284
16. Moghaddam, B., Pentland, A.: Face recognition using view-based and modular eigenspaces. Automatic Systems for the Identification and Inspection of Humans **SPIE Vol. 2277** (1994)
17. 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
18. Collobert, R., Bengio, S., Mariéthoz, J.: Torch: a modular machine learning software library. Technical Report IDIAP-RR 02-46, IDIAP (2002)
19. Rüping, S.: mySVM manual. Technical report, University of Dortmund, Lehrstuhl Informatik 8 (2000)