Face Recognition using Mixture-Distance and Raw Images

Steve Lawrence, Peter Yianilos, Ingemar Cox
{lawrence,pny,ingemar}@research.nj.nec.com
NEC Research, 4 Independence Way, Princeton, NJ 08540

Abstract

Earlier work suggests that *mixture-distance* can improve the performance of feature-based face recognition systems in which only a single training example is available for each individual. In this work we investigate the non-feature-based Eigenfaces technique of Turk and Pentland, replacing Euclidean distance with mixture-distance. In mixture-distance, a novel distance function is constructed based on local second-order statistics as estimated by modeling the training data with a mixture of normal densities. The approach is described and experimental results on a database of 600 people are presented, showing that mixture-distance can reduce the error rate by up to 73.9%. In the experimental setting considered, the results indicate that the simplest form of mixture distance yields considerable improvement. Additional, but less dramatic, improvement was possible with more complex forms. The results show that even in the absence of multiple training examples for each class, it is sometimes possible to infer an improved distance function from a statistical model of the training data. Therefore, researchers using Eigenfaces or similar pattern recognition techniques may find significant advantages by considering alternative distance metrics such as mixture-distance.

1 Introduction

Earlier work suggests that *mixture-distance* [9] can improve the performance of feature-based face recognition systems in which only a single training example is available for each individual [1]. In this work we investigate the non-feature-based[1] Eigenfaces technique of Turk and Pentland [8], replacing Euclidean distance with mixture-distance. In mixture-distance, a novel distance function is constructed based on local second-order statistics as estimated by modeling the training data with a mixture of normal densities.

In the limit of perfect normalization of the images, or infinite training data, no advantages would be gained in using alternative metrics to the standard Euclidean distance which is used in the standard Eigenfaces algorithm. However, real world datasets are always finite and normalization of face images is not perfect. In statistical terms, Euclidean distance may be viewed as corresponding to the assumption that each pattern vector is a class generator with unit covariance and mean coinciding with the pattern. Mixture-distance attempts to improve upon this assumption by considering the statistics of the training data. The training data is modeled as a mixture of normal densities, which is then used in a particular way to measure distance. We note that mixture models have also been used, in a different manner, for maximum likelihood detection of faces by Moghaddam and Pentland [3].

The remainder of this paper is organized as follows: section 2 summarizes the standard Eigenfaces algorithm, section 3 provides details of the dataset that we have used, section 4 summarizes the mixture-distance algorithm, section 5 presents and discusses the experimental methodology and results, and conclusions are given in section 6.

[1] Where "features" refers to the features used in feature-based face recognition methods such as the inter-iris distance. Eigenfaces could also be considered as features.

2 Eigenfaces

We used the Eigenfaces code from [7] which implements Eigenfaces as described in [8, 5, 4]. The procedure is as follows. Let the set of training images be $\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3, \ldots, \mathbf{T}_M$ where M is the number of training images. Each image is represented by a vector of length N, where N is the number of pixels in the image. The average face is defined as $\mathbf{A} = \frac{1}{M}\sum_{n=1}^{M}\mathbf{T}_n$. The faces differ from the average by $\mathbf{X}_i = \mathbf{T}_i - \mathbf{A}$. Eigenfaces seeks to use PCA to find a set of M orthonormal vectors, $\mathbf{u}_n$, which best describe the distribution of the data. The covariance matrix is:

$$\mathbf{C} = \frac{1}{M}\sum_{n=1}^{M}\mathbf{X}\mathbf{X}^T = \frac{1}{M}\mathbf{Y}\mathbf{Y}^T \qquad (1)$$

where $\mathbf{Y} = [\mathbf{X}_1\mathbf{X}_2\ldots\mathbf{X}_M]$. The matrix $\mathbf{C}$ is $N \times N$ and determining the N eigenvectors and eigenvalues is impractical for typical image sizes. However, if $M < N$, then there will only be M meaningful eigenvectors. These eigenvectors can be found by first finding the eigenvectors of an $M \times M$ matrix, and taking linear combinations of the face images. An $M \times M$ matrix $\mathbf{L} = \mathbf{Y}^T\mathbf{Y}$ is constructed where $L_{mn} = \mathbf{X}_m^T\mathbf{X}_n$. The M eigenvectors $\mathbf{v}_l$ are found for L. The eigenfaces are then computed with:

$$\mathbf{u}_l = \sum_{k=1}^{M}\mathbf{v}_{lk}\mathbf{X}_k, \qquad l = 1, \ldots, M \qquad (2)$$

These eigenfaces are ranked according to the associated eigenvalues and normalized by their norm. Faces are projected into "face space" using $\omega_k = \mathbf{u}_k^T(\mathbf{T} - \mathbf{A})$. The weights form a vector $\Omega^T = [\omega_1, \omega_2, \ldots, \omega_{M'}]$ which describes the contribution of each eigenface in representing the input image. These vectors may then be used in a standard pattern recognition algorithm such as the nearest neighbor algorithm using Euclidean distance.

3 Data

We have used a database of face images obtained from the MIT Media Lab. The database contains thousands of images which have been normalized for eye location and distance. We have further normalized these images using histogram equalization. These images differ from those used by Pentland et al. [2] in that Pentland et al. use extensive additional normalization for the geometry of the face, translation, lighting, contrast, rotation, and scale. We consider a more difficult task where such extensive normalization is not performed, e.g. due to computational requirements. Sample faces from the database are shown in figure 1.

4 Mixture-Distance

If many images of each person were available, then each person could be considered a pattern class, and the methods of statistical pattern recognition could be used to build a model for each person. A common approach models each class as a normal density, so for each person there is a corresponding mean feature vector and covariance matrix. The probability of an unknown pattern conditioned on each model is then easily computed. Using a prior distribution on the individuals in the database (flat for example) the classification task is completed in the standard Bayesian fashion by computing the a posteriori probability of each person, conditioned on observation of the query. If the computation is performed using log probabilities, it is slightly less expensive computationally and the distance metric is the well known Mahalanobis distance. However, in practice it is not uncommon to have only a single training example for each person, as is the case here.

Mixture-distance employs a mixture model G, which is a collection of probability models, $G_1, G_2, \ldots, G_n$ and non-negative mixing param-

Figure 1. Sample faces from the MIT database.

eters $c_1, c_2, \ldots, c_n$ summing to unity, such that:

$$G(x) = \sum_{k=1}^{n} c_k G_k(x) \quad (3)$$

The elements G_i of G are Gaussian so that G is a Gaussian or normal mixture. The task of estimating the parameters of a normal mixture model has been extensively studied. We have used the well known expectation-maximization algorithm (EM) [6]. We initialize the mean μ_i and the covariance Σ_i for each Gaussian G_i using the K-means algorithm. The determinants of the covariance matrices and the inverse covariance matrices have to be computed at each iteration of the EM algorithm to evaluate the new value of the log likelihood. However, the covariance matrices are often ill-conditioned. This problem has been addressed by multiplying the off-diagonal elements of the covariance matrices by a number, $0 \leq f < 1$, to reduce their influence. This factor can be seen to select between first-order ($f = 0$) and second-order ($f = 1$) models.

Two methods have been used to classify new feature vectors, as in [1]. In *"hard"* vector quantization (hard VQ), each training example is assigned to one Gaussian: Y_i is assigned to the Gaussian G_l where $l = \text{argmax}_j \ P(G_j|Y_i)$. When attempting to classify a query Q, the Mahalanobis distance between Q and each training example Y_i is computed as follows:

$$\text{distance}(Q, Y_i) = (Q - Y_i)^T {\Sigma_l}^{-1} (Q - Y_i) \quad (4)$$

where Σ_l is the covariance matrix of the Gaussian to which Y_i has been assigned. Q is classified as the point Y_k where $k = \text{argmin}_i \, \text{distance}(Q, Y_i)$.

In *"soft"* vector quantization (soft VQ), instead of assigning each training example to one Gaussian, we consider that a point can belong to several Gaussians at the same time and can therefore be described by the properties of these Gaussians. A query Q is classified using point Y_l where $l = \text{argmax}_i \ P(Y_i|Q)$, where

$$P(Y_i|Q) = \frac{P(Q|Y_i)P(Y_i)}{P(Q)} = \frac{P(Q|Y_i)P(Y_i)}{\sum_{k=1}^{N} P(Q|Y_k)P(Y_k)} \quad (5)$$

where

$$P(Q|Y_i) = \sum_{j=1}^{M} P(Q|G_j, Y_i)P(G_j|Y_i) \quad (6)$$

$P(G_j|Y_i)$ is the posterior probability and $P(Q|G_j, Y_i)$ is computed according to the method described in [9]:

$$P(Q|G_j, Y_i) = \frac{1}{(2\Pi)^{\frac{d}{2}} |\Sigma_j|^{\frac{1}{2}}} \exp -\frac{1}{2}(Q - Y_i)^T {\Sigma_j}^{-1} (Q - Y_i) \quad (7)$$

See [1] for intuition behind the formulation of mixture-distance.

5 Results

In our experiments we used 600 training images and 131 test images. Because no class labels were available, the test images were chosen manually by

Figure 2. The average face and the first five eigenfaces for a sample 600 class training set.

scanning the database for duplicate images of the same person. This procedure is expected to lead to a biased test set due to the fact that the judgment of one human was used to determine which images were of the same person, and that mistakes are possible, e.g. additional images of the same person may be present in the training set yet labelled as different classes. For each set of two images that were identified to be of the same person, one was used in the training set and the other in the test set. The training set was further augmented with 469 images which were randomly selected from the database. This random selection was repeated five times to create five different training sets. All experiments below were done over these five training sets. The eigenfaces were recalculated for each training set. Figure 2 shows the average face and the first five eigenfaces for a sample 600 class training set. We note that in comparison to the eigenface images presented by [3], the eigenfaces shown appear to reflect more variability in the face images with respect to face size and location, as might be expected due to the less precise normalization.

We used mixture models with from 1 to 5 Gaussians and a *flat mixture of mixtures* model which consisted of a combination of the 5 individual mixture models. We tested both the hard VQ and soft VQ versions of mixture-distance. As was also found in [1], we found that these provided similar performance. We therefore show only the hard VQ results in this paper, and note that the hard VQ algorithm is less computationally expensive.

Figure 3 shows the results using 10 eigenfaces and figure 4 shows the results using 30 eigenfaces. It can be seen that performance is typically better with higher f, where f is the off-diagonal scaling factor used to select between first and second-order models. Apart from $f = 0$, the use of mixture distance significantly improves performance over the standard Eigenfaces algorithm. However, it can be seen that using only one Gaussian provides significant improvement. Using greater than one Gaussian can be seen to be beneficial in most cases, in particular for the 10 eigenfaces case. The mixture of mixtures models perform comparably to the best of the individual models, providing a solution to the model selection problem.

For the 10 eigenfaces case, t-tests indicate that the (MD-1 (mixture-distance, 1 Gaussian), MD-2, MD-Mixture (mixture-distance, mixture of models with 1 to 5 Gaussians)) models have different means in comparison to the standard Eigenfaces case at p values[2] of (0.019, 0.0013, and 0.0007) for $f = 0.5$ and p values of (0.0044, 0.00011, 0.00065) for $f = 0.99$. t-tests comparing the MD-1 and MD-2 cases produce p values of 0.024 and 0.025 for $f = 0.5$ and $f = 0.99$ respectively. t-tests comparing the MD-1 and MD-Mixture cases produce p values of 0.32 and 0.0042 for $f = 0.5$ and $f = 0.99$ respectively. The percentage reduction in error for the $f = 0.99$ case with respect to the standard Eigenfaces algorithm was 31.3%, 44.4%, and 42% respectively for the MD-1, MD-2, and MD-Mixture models.

For the 30 eigenfaces case, t-tests indicate that the (MD-1, MD-2, MD-Mixture) models have different means in comparison to the standard Eigenfaces case at p values of (7.2×10^{-6}, 5.7×10^{-5},

[2] p is the probability of the two samples coming from the same distribution. Low values of p indicate a high probability that the differences are significant.

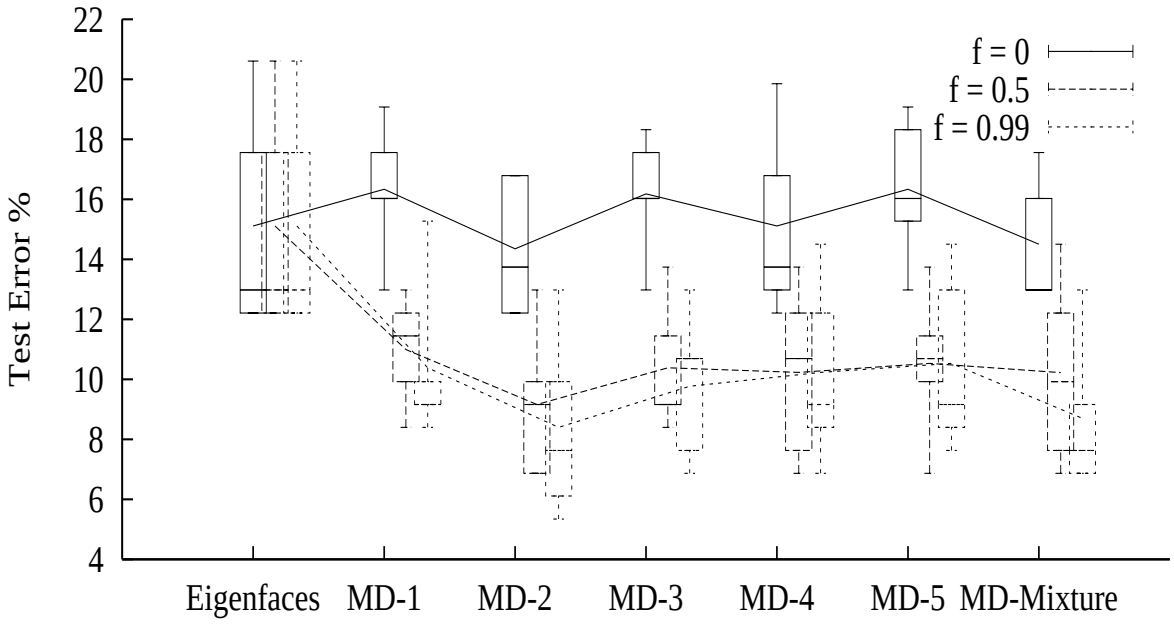

Figure 3. Results for 10 eigenfaces shown using box-whiskers plots. Left to right: standard Eigenfaces, mixture-distance with 1 to 5 Gaussians, and mixture-distance with a mixture of the models using 1 to 5 Gaussians. Results are shown for $f = 0$, 0.5, and 0.99 where f is the scaling factor which selects between first-order ($f = 0$) and second-order ($f = 1$) models.

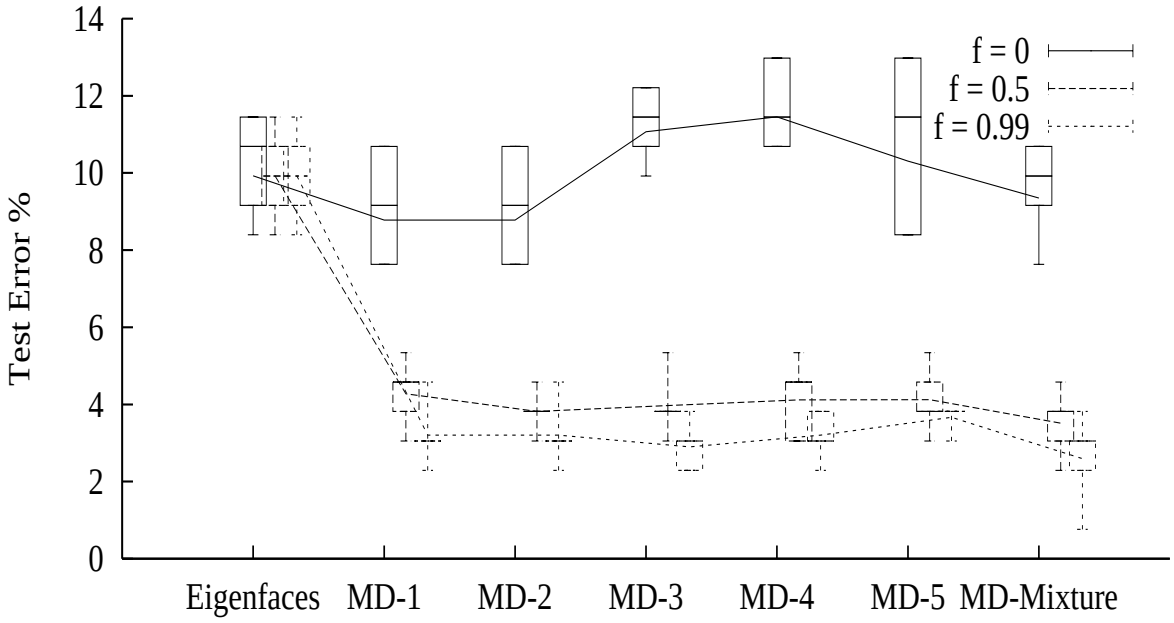

Figure 4. Results for 30 eigenfaces shown using box-whiskers plots. Left to right: standard Eigenfaces, mixture-distance with 1 to 5 Gaussians, and mixture-distance with a mixture of the models using 1 to 5 Gaussians. Results are shown for $f = 0$, 0.5, and 0.99 where f is the scaling factor which selects between first-order ($f = 0$) and second-order ($f = 1$) models.

and 4.3×10^{-6}) for $f = 0.5$ and p values of (1.9×10^{-5}, 1.9×10^{-5}, and 2.6×10^{-6}) for $f = 0.99$. t-tests comparing the MD-1 and MD-2 cases produce p values of 0.07 and 1.0 for $f = 0.5$ and $f = 0.99$ respectively. t-tests comparing the MD-1 and MD-Mixture cases produce p values of 2.8×10^{-8} and 0.099 for $f = 0.5$ and $f = 0.99$ respectively. The percentage reduction in error for the $f = 0.99$ case with respect to the standard Eigenfaces algorithm was 66.5%, 66.5%, and 73.9% respectively for the MD-1, MD-2, and MD-Mixture models.

6 Conclusions

Earlier work has suggested that *mixture-distance* can improve the performance of feature-based face recognition systems in which only a single training example is available for each individual. In this work we investigated replacing Euclidean distance with mixture-distance in Eigenfaces. We found that mixture-distance resulted in significant reduction in the error rate, up to 73.9%. In the experimental setting considered, the simplest form of mixture-distance yielded considerable improvement. Additional, but less dramatic, improvement was possible with more complex forms. *Flat mixture of mixtures* models were found to perform comparably to the best of the individual models, providing a solution to the model selection problem. The results show that even in the absence of multiple training examples for each class, it is sometimes possible to infer an improved distance function from a statistical model of the training data. Therefore, researchers using Eigenfaces or similar pattern recognition techniques may find significant advantages by considering alternative distance metrics such as mixture-distance.

Acknowledgments

We would like to thank the MIT Media Lab for providing the face images.

References

[1] Ingemar J. Cox, Joumana Ghosn, and Peter N. Yianilos. Feature-based face recognition using mixture-distance. In *International Conference on Computer Vision and Pattern Recognition*, pages 209–216. IEEE Press, 1996.

[2] B. Moghaddam and A. Pentland. Face recognition using view-based and modular eigenspaces. In *Automatic Systems for the Identification and Inspection of Humans, SPIE*, volume 2257, 1994.

[3] B. Moghaddam and A. Pentland. Probabilistic visual learning for object detection. In *International Conference on Computer Vision*, pages 786–793, 1995.

[4] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *IEEE Conference on Computer Vision and Pattern Recognition,* 1994.

[5] A. Pentland, T. Starner, N. Etcoff, A. Masoiu, O. Oliyide, and M. Turk. Experiments with Eigenfaces. In *Looking at People Workshop, International Joint Conference on Artificial Intelligence 1993*, Chamberry, France, 1993.

[6] R.A. Redner and H.F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26:195–239, 1984.

[7] Thad Starner. Eigenfaces code, 1997.

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

[9] Peter Yianilos. Metric learning via normal mixtures. Technical report, NEC Research Institute, 1995.