FACE RECOGNITION USING KERNEL EIGENFACES

Ming-Hsuan Yang Narendra Ahuja David Kriegman

Department of Computer Science and Beckman Institute
University of Illinois at Urbana-Champaign, Urbana, IL 61801
E-mail: {myang1, n-ahuja, kriegman}@uiuc.edu
Web Page: http://vision.ai.uiuc.edu

ABSTRACT

Eigenface or Principal Component Analysis (PCA) methods have demonstrated their success in face recognition, detection, and tracking. The representation in PCA is based on the second order statistics of the image set, and does not address higher order statistical dependencies such as the relationships among three or more pixels. Recently Higher Order Statistics (HOS) have been used as a more informative low dimensional representation than PCA for face and vehicle detection. In this paper we investigate a generalization of PCA, Kernel Principal Component Analysis (Kernel PCA), for learning low dimensional representations in the context of face recognition. In contrast to HOS, Kernel PCA computes the higher order statistics without the combinatorial explosion of time and memory complexity. While PCA aims to find a second order correlation of patterns, Kernel PCA provides a replacement which takes into account higher order correlations. We compare the recognition results using kernel methods with Eigenface methods on two benchmarks. Empirical results show that Kernel PCA outperforms the Eigenface method in face recognition.

1. MOTIVATION AND APPROACH

Subspace methods have been applied successfully in applications such as face recognition using Eigenfaces (or PCA face) [11] [5], face detection [5], object recognition [6], and tracking [1]. Representations such as PCA encode the pattern information based on second order dependencies, i.e., pixelwise covariance among the pixels, and are insensitive to the dependencies of multiple (more than two) pixels in the patterns. Since the eigenvectors in PCA are the orthonormal bases, the principal components are uncorrelated. In other words, the coefficients for one of the axes cannot be linearly represented from the coefficients of the other axes.

Higher order dependencies in an image include nonlinear relations among the pixel intensity values, such as the relationships among three or more pixels in an edge or a curve, which can capture important information for recognition. Several researchers have conjectured that higher order statistics may be crucial to better represent complex patterns. Recently, Higher Order Statistics (HOS) have been applied to visual learning problems. Rajagopalan et al. use HOS of the images of a target object to get a better approximation of an unknown distribution. Experiments on face detection [7] and vehicle detection [8] show comparable, if no better, results than other PCA-based methods.

HOS usually works by projecting the input patterns to a higher dimensional space R^F before computing the cumulants. The k-th order cumulant is defined in terms of its joint moments of order up to k. For zero mean random variables x_1, x_2, x_3, x_4, the second, third and fourth order cumulants are given by

$$\begin{aligned} Cum(x_1,x_2) &= E[x_1x_2] \\ Cum(x_1,x_2,x_3) &= E[x_1x_2x_3] \\ Cum(x_1,x_2,x_3,x_4) &= E[x_1x_2x_3x_4] - E[x_1x_2]E[x_3x_4] - \\ & \quad E[x_1x_3]E[x_2x_4] - E[x_1x_4]E[x_2x_3] \end{aligned}$$

Note the computation involved in HOS depends on the order of cumulants and is usually heavy because of computing expectations in a high dimensional space.

In contrast to computing cumulants in HOS, we seek a formulation which computes the higher order statistics using only dot products, $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$, of the training patterns $\mathbf{x}$ where Φ is a nonlinear projection function. Since we can compute these dot products efficiently, we can solve the original problem without explicitly mapping to R^F. This is achieved using Mercer kernels where a kernel $k(\mathbf{x}_i, \mathbf{x}_j)$ computes the dot product in some feature space R^F, i.e., $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$.

The idea of using kernel methods has also been adopted in the Support Vector Machines (SVMs) in which kernel functions replace the nonlinear projection functions such that an optimal separating hyperplane can be constructed efficiently [2]. Schölkopf et al. proposed the use of Kernel PCA for object recognition in which the principal components of an object image comprise a feature vector to train a SVM [10]. Empirical results on character recognition using MNIST data set and object recondition using MPI chair database

0-7803-6297-7/00/$10.00 © 2000 IEEE

show that Kernel PCA is able to extract nonlinear features. Since much of the important information may be contained in the high order relationships among the pixels of a face image, we investigate the use of Kernel PCA for face recognition and compare its performance against the Eigenface method.

2. KERNEL PRINCIPAL COMPONENT ANALYSIS

Given a set of zero-mean observations $\mathbf{x}_k, k = 1, \ldots, M$, $\mathbf{x}_k \in R^N$, and $\sum_{k=1}^{M} \mathbf{x}_k = 0$, the covariance matrix is

$$C = \frac{1}{M} \sum_{j=1}^{M} \mathbf{x}_j \mathbf{x}_j^T \qquad (1)$$

PCA aims to find the projection direction that maximizes the variance, which is equivalent to finding the eigenvalue from the covariance matrix

$$\lambda \mathbf{w} = C\mathbf{w} \qquad (2)$$

for eigenvalues $\lambda \geq 0$ and $\mathbf{w} \in R^N$. Since $C\mathbf{w} = \frac{1}{M}\sum_{j=1}^{M}(\mathbf{x}_j \cdot \mathbf{w})\mathbf{x}_j$, all solutions $\mathbf{w}$ with $\lambda \neq 0$ must lie in the span of $\mathbf{x}_1, \ldots, \mathbf{x}_M$. Therefore

$$\lambda(\mathbf{x}_k \cdot \mathbf{w}) = (\mathbf{x}_k \cdot C\mathbf{w}),\ k = 1, \ldots, M \qquad (3)$$

In Kernel PCA, each vector $\mathbf{x}$ is projected from the input space, R^N, to a high dimensional feature space, R^F, by a nonlinear map:

$$\Phi : R^N \rightarrow R^F, F \gg N \qquad (4)$$

Note that the dimensionality of the feature space can be arbitrarily large. In R^F, the covariance matrix of $\Phi(\mathbf{x})$ is

$$C^{\Phi} = \frac{1}{M} \sum_{j=1}^{M} \Phi(\mathbf{x}_j)\Phi(\mathbf{x}_j)^T \qquad (5)$$

and the corresponding eigenvalue problem is

$$\lambda \mathbf{w}^{\Phi} = C\mathbf{w}^{\Phi} \qquad (6)$$

All solutions $\mathbf{w}^{\Phi}$ with $\lambda \neq 0$ lie in the span of $\Phi(\mathbf{x}_1)$, ..., $\Phi(\mathbf{x}_M)$.

$$\lambda(\Phi(\mathbf{x}_k) \cdot \mathbf{w}^{\Phi}) = (\Phi(\mathbf{x}_k) \cdot C\mathbf{w}^{\Phi}) \quad k = 1, \ldots, M \qquad (7)$$

and $\mathbf{w}^{\Phi}$ lie in the span of $\Phi(\mathbf{x}_1)$, ..., $\Phi(\mathbf{x}_M)$ such that

$$\mathbf{w}^{\Phi} = \sum_{i=1}^{M} \alpha_i \Phi(\mathbf{x}_i) \qquad (8)$$

Using Equations (7) and (8), we have, for $k = 1, \ldots, M$,

$$\lambda \textstyle\sum_{i=1}^{M} \alpha_i(\Phi(\mathbf{x}_k) \cdot \Phi(\mathbf{x}_i)) = \\ \frac{1}{M} \textstyle\sum_{i=1}^{M} \alpha_i(\Phi(\mathbf{x}_k) \cdot \sum_{j=1}^{M} \Phi(\mathbf{x}_j))(\Phi(\mathbf{x}_j) \cdot \Phi(\mathbf{x}_i)) \qquad (9)$$

Defining an $M \times M$ matrix K by

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j) \qquad (10)$$

We can rewrite Equation (9) as

$$M\lambda K\boldsymbol{\alpha} = K^2\boldsymbol{\alpha} \qquad (11)$$

where $\boldsymbol{\alpha}$ denotes a column vector with entries $\alpha_1, \ldots, \alpha_M$. The solutions of Equation (11) is the same to the following eigenvalue problem,

$$M\lambda\boldsymbol{\alpha} = K\boldsymbol{\alpha} \qquad (12)$$

See [9] [10] for technical details on the equivalence of these two problems and how to center the vectors $\Phi(\mathbf{x})$ in R^F.

Boser, Guyon and Vapnik suggested the use of Gaussian Radial Basis Function kernel [2]

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}) \qquad (13)$$

In this paper, we use the polynomial kernel of degree d for the sake of computational efficiency, i.e.,

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d \qquad (14)$$

Note that conventional PCA is a special case of Kernel PCA with polynomial kernel of first order. In other words, Kernel PCA is a generalization of conventional PCA since different kernels can be utilized for different nonlinear projections. Table 1 summarizes the procedure for Kernel PCA.

We can now project the vectors in R^F to a lower dimensional space spanned by the eigenvectors $\mathbf{w}^{\Phi}$, Let $\mathbf{x}$ be a test sample whose projection is $\Phi(\mathbf{x})$ in R^F, then the projection of $\Phi(\mathbf{x})$ onto the eigenvectors $\mathbf{w}^{\Phi}$ are the nonlinear principal components corresponding to Φ:

$$\mathbf{w}^{\Phi} \cdot \Phi(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i(\Phi(\mathbf{x}_i) \cdot \Phi(x)) = \sum_{i=1}^{M} \alpha_i k(\mathbf{x}_i, \mathbf{x}) \qquad (15)$$

In other words, we can extract the first q $(1 \leq q \leq M)$ nonlinear principal components using the kernel function without the expensive operation to explicitly project the patterns to a high dimensional space R^F. The first q components correspond to the first q nonincreasing eigenvalues of Equation (12).

Table 1: Kernel PCA

1. Compute the matrix $K_{ij} = (k(\mathbf{x}_i, \mathbf{x}_j))_{ij}$
2. Solve

$$M\lambda\boldsymbol{\alpha} = K\boldsymbol{\alpha}$$

   and normalize the eigenvector expansion coefficients. Let $\lambda_1 \leq \lambda_2 \leq \cdot \leq \lambda_M$ denote the eigenvalues of K and $\boldsymbol{\alpha}^1, \dots, \boldsymbol{\alpha}^M$ the corresponding eigenvectors, with λ_p be the first nonzero eigenvalue. Normalize the coefficients by requiring $\lambda_i(\boldsymbol{\alpha}^i \cdot \boldsymbol{\alpha}^i) = 1$ for $p \leq i \leq M$.
3. Extract the principal components (corresponding to the kernel k) of the test point $\mathbf{x}$, and compute the projections onto the eigenvectors by

$$(\mathbf{w}^n \cdot \Phi(\mathbf{x})) = \sum_{i=1}^{M} \alpha_i^n k(\mathbf{x}_i, \mathbf{x})$$

3. PROPERTIES OF KERNEL PCA

We discuss sevearl properties of Kernel PCA in terms of feature extraction and reconstruction in this section.

3.1. Dimensionality and Feature Extraction

Kernel PCA method can extract more principal components than linear PCA. Consider a problem consisting of M observations $\mathbf{x}$ where the dimension of $\mathbf{x}$ is N and $M \gg N$. Linear PCA can find at most N nonzero eigenvalues from the covariance matrix C $(C = \frac{1}{M}\sum i = 1^M \mathbf{x}_i\mathbf{x}_i^T$. In contrast, Kernel PCA can find up to M nonzero eigenvalues from the covariance matrix $C^\Phi = \frac{1}{M}\sum_{i=1}^{M} \Phi(\mathbf{x}_i)\Phi(\mathbf{x}_i^T)$ where Φ is a nonlinear mapping function that can project $\mathbf{x}_i$ to a possibly infinite-dimensional feature space.

3.2. Reconstruction

Since PCA is essentially a basis transformation, each pattern can be exactly reconstructed using all the principal components and the basis vectors (i.e., eigenvectors).

In contrast, there is no direct counterpart in Kernel PCA. Due to nonlinear mapping, a vector in high dimensional feature space does not necessarily have a pre-image in the input space. We can at best find an approximate reconstruction of the image of a pattern in R^F from its nonlinear components. This can be achieved by a regression method for estimating the mapping from kernel-based principal components to the input space.

4. EXPERIMENTS

We tested Kernel PCA with polynomial kernels against conventional PCA using two image databases. The Yale database contains 165 images of 11 subjects that includes variation in both facial expression and lighting. For efficiency, each image has been downsampled to 29 × 41 pixels. Figure 1 shows 22 closely cropped images which include internal facial structures such as the eyebrow, eyes, nose, mouth and chin, but do not contain the facial contours.

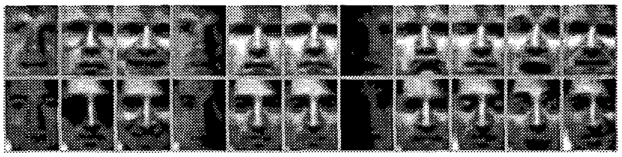

Figure 1: The Yale database contains 165 frontal face images of 15 individuals taken with variation both in facial expression and lighting.

The experiments were performed using the "leave-one-out" strategy: To classify an image of person, that image is removed from the training set of $M - 1$ images and the dimensionality reduction matrix $\mathbf{w}^\Phi$ is computed. All the M images in the training set are projected to a reduced space using the computed matrix $\mathbf{w}^\Phi$ and recognition is performed using a nearest neighbor classification. The number of eigenvectors (or principal components) are empirically determined to achieve lowest error rate by each method. Table 2 shows the experimental results. Empirical results show that Kernel PCA method with a cubic polynomial kernel achieve the lowest error rate. Furthermore, the results show that Kernel PCA methods are insensitive to the degree of polynomial kernels.

Table 2: Experimental results on Yale database

Method	Reduced space	Error Rate (%)
Eigenface	40	28.49
Kernel PCA, d=2	80	27.27
Kernel PCA, d=3	60	24.24
Kernel PCA, d=4	60	24.85
Kernel PCA, d=10	50	26.01

The AT&T (formerly Olivetti) database contains 400 images of 40 subjects that include variation in facial expression and pose. Each face image is downsampled to 23 × 28 to reduce the computational complexity. Figure 2 shows images of two subjects. In contrast to the Yale database, the images include the facial contours and certain pose variations. However, the lighting conditions remain the same. Figure 2 shows some sample images.

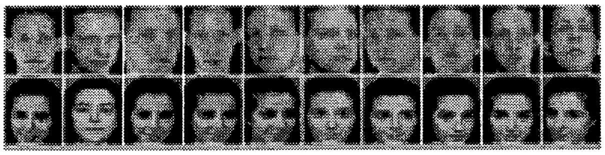

Figure 2: The AT&T (formerly Olivetti) database contains 400 frontal face images of 40 subjects with variation in facial expression and pose.

We use the same strategy with the experiments using the Yale data set. Table 3 summarizes the empirical results. Consistent with the experiments on Yale database, Kernel PCA methods achieve lower error rates than the Eigenface approach on the AT&T dataset.

Table 3: Experimental results on AT&T database

Method	Reduced space	Error Rate (%)
Eigenface	30	2.75
Kernel PCA, d=2	50	2.50
Kernel PCA, d=3	50	2.00
Kernel PCA, d=4	60	2.25
Kernel PCA, d=10	80	2.25

5. DISCUSSION AND CONCLUSION

The representation in the Eigenface approaches is based on the second order statistics of the image set, i.e., covariance matrix, and does not use high order statistical dependencies such as the relationships among three or more pixels. In a task such as face recognition, much of the important information may be contained in the high order statistical relationships among the pixels. We have investigated Kernel PCA and demonstrated that it provides a more effective representation for face recognition for face recognition. Compared to other techniques for nonlinear feature extraction, Kernel PCA has the advantages that it does not require nonlinear optimization, but only the solution of an eigenvalue problem. Experimental results on two benchmark databases show that Kernel PCA method achieves a lower error rate than the Eigenface approach in face recognition.

Future research will focus on analyzing face recognition methods using other kernel methods in high dimensional space. We plan to investigate and compare the performance of face recognition methods using Kernel Fisher Linear Discriminant [4], Independent Component Analysis [3] and Kernel PCA.

6. REFERENCES

[1] M. J. Black and A. Jepson. Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. *International Journal of Computer Vision*, 26(1):63–84, 1996.

[2] B. E. Boser, I. M. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, pages 144–152, 1992.

[3] G. Donato, M. S. Bartlett, J. C. Hager, P. Ekman, and T. J. Sejnowski. Classifying facial actions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):974–989, 1999.

[4] S. Mika, G. Rätsch, J. Weston, and B. Schölkopf. Fisher discriminant analysis with kernels. In *Proceedings of IEEE Neural Networks for Signal Processing Workshop*, 1999.

[5] B. Moghaddam and A. Pentland. Probabilistic visual learning for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):696–710, 1997.

[6] H. Murase and S. Nayar. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[7] A. Rajagopalan, K. Kumar, J. Karlekar, R. Manivasakan, M. Patil, U. Desai, P. Poonacha, and S. Chaudhuri. Finding faces in photographs. In *Proceedings of the Sixth International Conference on Computer Vision*, pages 640–645, 1998.

[8] A. N. Rajagopalan, P. Burlina, and R. Chellappa. Higher order statistical learning for vehicle detection in images. In *Proceedings of the Seventh International Conference on Computer Vision*, volume 2, pages 1204–1209, 1999.

[9] B. Schölkopf. *Support Vector Learning.* PhD thesis, Informatik der Technischen Universitat Berlin, 1997.

[10] B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[11] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.