Kernel Methods for Nonlinear Discriminative Data Analysis

Xiuwen Liu[1,⋆] and Washington Mio[2]

[1] Department of Computer Science, Florida State University, Tallahassee, 32306
Phone: (850) 644-0050, Fax: (850) 644-0058
liux@cs.fsu.edu
[2] Department of Mathematics, Florida State University, Tallahassee, FL 32306

Abstract. Optimal Component Analysis (OCA) is a linear subspace technique for dimensionality reduction designed to optimize object classification and recognition performance. The linear nature of OCA often limits recognition performance, if the underlying data structure is nonlinear or cluster structures are complex. To address these problems, we investigate a kernel analogue of OCA, which consists of applying OCA techniques to the data after it has been mapped nonlinearly into a new feature space, typically a high (possibly infinite) dimensional Hilbert space. In this paper, we study both the theoretical and algorithmic aspects of the problem and report results obtained in several object recognition experiments.

1 Introduction

Modeling nonlinearity in observed data for tasks such as dimensionality reduction in data representation for efficient classification and recognition of objects and patterns is a problem that arises in numerous contexts. For example, in image-based object recognition, nonlinearity often arises as a result of varying poses, illumination and other factors. Kernel methods have been widely used as a general strategy for simplifying data structure so that it becomes amenable to linear methods. One typically maps a given dataset in Euclidean space $\mathbb{R}^n$ nonlinearly into a very high (possibly infinite) dimensional Hilbert space $\mathbb{H}$ and analyzes the transformed data with more standard techniques. Such methods have been investigated in the context of support vector machines [12], principal component analysis [10], independent component analysis [1], and Fisher discriminant analysis [2]. For practical feasibility, the usual assumption is that the nonlinear map $\Phi\colon \mathbb{R}^n \to \mathbb{H}$ is not known explicitly, only the relative positions of the points $\Phi(x)$, $x \in \mathbb{R}^n$, given by the inner products $k(x,y) = \Phi(x) \cdot \Phi(y)$, $x, y \in \mathbb{R}^n$. The function $k(x,y)$ is referred to as a *kernel function*. Thus, dimension reduction techniques and classifiers should only require knowledge of the kernel k, not the function Φ.

In this paper, we present a kernel analogue of a linear subspace technique developed by Liu et al. in [7, 11] that has been termed Optimal Component Analysis (OCA). Given training data for a specific classification problem, OCA is a technique for finding an

⋆ Corresponding author.

A. Rangarajan et al. (Eds.): EMMCVPR 2005, LNCS 3757, pp. 584–599, 2005.

optimal subspace of feature space for dimensionality reduction for classification and recognition. Although originally developed in the context of images for the nearest-neighbor classifier, the method applies to more general data classification based on other criteria, as well. We address both the theoretical and computational aspects of kernel OCA. For the methodology to be useful in practice, it is crucial that we develop an algorithmic approach that leads to effective computational tools. This will be achieved by exploiting the differential geometric properties of Grassmann manifolds [5, 13], as discussed in more detail below. Several object recognition experiments will illustrate the fact that high recognition rates can be achieved with kernel OCA in a computationally efficient manner. A special case of kernel OCA was studied in [14], where the kernel function is required to satisfy the additional constraint that Φ preserves orthonormality. Under this assumption, the problem can be more easily reduced to OCA in Euclidean space.

The paper is organized as follows. In Sect. 2, we give a brief overview of optimal component analysis in Euclidean space, which is followed by a formulation of the corresponding problem in kernel space in Sect. 3. Sect. 4 shows how an efficient stochastic gradient search algorithm can be devised by exploiting a special representation of elements of Grassmann manifolds. Sect. 5 contains a systematic set of experiments and Sect. 6 concludes the paper with a discussion of future research.

2 Optimal Component Analysis

We begin with a brief review of Optimal Component Analysis (OCA) in Euclidean space $\mathbb{R}^m$. Suppose that a given dataset is divided in *training* and *validation* sets, each consisting of representatives of P different classes of objects. For $1 \leq c \leq P$, we denote by $x_{c,1}, \ldots, x_{c,t_c}$ and $y_{c,1}, \ldots, y_{c,v_c}$ the elements in the training and validation sets, resp., that belong to class c. Given an r-dimensional subspace U of $\mathbb{R}^m$ and $x, y \in \mathbb{R}^m$, we let $d(x, y; U)$ denote the distance between the orthogonal projections of x and y onto U. The quantity

$$\rho(y_{c,i}; U) = \frac{\min_{c \neq b,j} d^2(y_{c,i}, x_{b,j}; U)}{\min_j d^2(y_{c,i}, x_{c,j}; U) + \epsilon} \tag{1}$$

measures how well the *nearest-neighbor classifier* applied to the data projected onto U identifies the element $y_{c,i}$ as belonging to class c; a large value $\rho(y_{c,i}; U)$ indicates that, after projection, $y_{c,i}$ is much closer to the class it belongs than to other classes. Here, $\epsilon > 0$ is a small number used to prevent vanishing denominators. The function ρ is a mild variant of that used in [7], with the distance d squared to ensure smoothness. Note that (1) can be modified to reflect the performance of a more general K-nearest-neighbor classifier. Define a performance function by

$$F(U) = \frac{1}{P} \sum_{c=1}^{P} \left(\frac{1}{v_c} \sum_{i=1}^{v_c} h\left(\rho(y_{c,i} : U) - 1\right) \right), \tag{2}$$

where h is a monotonically increasing bounded function. A common choice is

$$h(x) = \frac{1}{1 + e^{-2\beta x}},$$

for which the limit value of $F(U)$, as $\beta \to \infty$, is precisely the recognition performance of the nearest-neighbor classifier after projection to the subspace U. Unlike the actual recognition performance, $F(U)$ is smooth so that we can approach the search for its maxima using gradient-type algorithms. The function h is used to control bias with respect to particular classes in measurements of performance.

Let $\mathcal{G}(m,r)$ be the Grassmann manifold [5, 13] of r-planes in $\mathbb{R}^m$. An optimal r-dimensional subspace for the given classification problem from the viewpoint of the available data is given by

$$\hat{U} = \operatorname*{argmax}_{U \in \mathcal{G}_{m,r}} F(U).$$

An algorithmic procedure for estimating $\hat{U}$ on $\mathcal{G}(m,r)$ using a stochastic gradient search is described in [7]. Notice that, in practice, for this approach to classification and recognition to be feasible, the estimation of the gradient of F must be carried out efficiently.

3 Subspace Representation

In data analysis using kernel methods, one typically maps a given set of data $x_1, \ldots, x_M$ in $\mathbb{R}^n$ to a Hilbert space $(\mathbb{H}, \langle\, , \,\rangle)$ using a nonlinear map $\Phi\colon \mathbb{R}^n \to \mathbb{H}$, and then applies linear subspace techniques to the collection $\Phi(x_1), \ldots, \Phi(x_M)$. The typical assumption is that Φ is not known explicitly, only the kernel function $k(x,y) = \Phi(x) \cdot \Phi(y)$. The problem of determining what functions $k(x,y)$ are kernels associated with a mapping Φ has been studied in [4, 12, 10]. Some of the most commonly used kernel functions are

$$k(x,y) = (x \cdot y)^d,$$

which corresponds to mapping $\mathbb{R}^n$ into a higher dimensional space using all monomials of order d in the input variables [9], and the *Gaussian kernel*

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right).$$

We shall adapt OCA to this setting, by projecting vectors of the form $\Phi(x)$, $x \in \mathbb{R}^n$, onto subspaces of

$$V = \operatorname{span}\{\Phi(x_1), \ldots, \Phi(x_M)\} \subseteq \mathbb{H},$$

where we use the nearest-neighbor (or more generally K-nearest-neighbor) criterion for classification and recognition. For this purpose, we must be able to measure distances between projected vectors solely in terms of the kernel function.

Remarks.

(a) Note that OCA, in its original formulation, does not restrict subspaces to the span of the data points. This is an important difference and philosophically reflects the fact that, in the kernel approach, cluster structures are expected to be simplified by applying a non-linear map Φ to the original data, so that high recognition rates can be achieved

only using projections to low-dimensional subspaces of the span of the kernelized data. This will lead to significant gains in computational efficiency.

(b) If the need arises, one can allow other subspaces of the Hilbert space $\mathbb{H}$, for example, by taking a set of vectors $\{\bar{x}_1, \ldots, \bar{x}_N\} \subset \mathbb{R}^n$ and replacing V above with

$$V = \text{span}\,\{\Phi(\bar{x}_1), \ldots, \Phi(\bar{x}_N)\}\,.$$

Thus, the formulation given here is not limited to the span of the training images. This raises the problems of learning and selecting $\{\bar{x}_1, \ldots, \bar{x}_N\}$ either from data or based on associated physical processes; these issues require further investigation.

Each element $a = (a_1, \ldots, a_M)^T \in \mathbb{R}^{M\times 1}$ defines a vector $v \in V$ given by $v = \sum_{i=1}^{M} a_i \Phi(x_i)$. Form the symmetric Gram matrix $K \in \mathbb{R}^{M\times M}$, whose entries are

$$K_{ij} = \Phi(x_i) \cdot \Phi(x_j)\,.$$

If $a, b \in \mathbb{R}^{M\times 1}$ represent $v, w \in V$, then

$$\langle v, w\rangle = a^T K b. \tag{3}$$

Our first goal is to find an orthonormal basis of V in the a-representation. For this, we diagonalize the Gram matrix, and let $d_j^* = (d_{1j}^*, \ldots, d_{Mj}^*)^T$, $1 \le j \le m$, be an orthonormal set (with respect to the standard inner product on $\mathbb{R}^M$) associated with the nonzero eigenvalues $\lambda_1, \ldots, \lambda_m$ of K, where $m = \dim V = \text{rank}\,K$. It follows from (3) that

$$d_1 = d_1^*/\sqrt{\lambda_1}\ ,\ \ldots\ ,\ d_m = d_m^*/\sqrt{\lambda_m} \tag{4}$$

represent an orthonormal basis of V. This fact can be expressed as $D^T K D = I_m$, where D is the $M \times m$ matrix whose columns are $d_1, \ldots, d_m$. Note that D can be constructed as indicated since the Gram matrix K is positive semi-definite. In addition, one can choose a subset of the eigenvectors (with nonzero eigenvalues) to further reduce computational costs if needed.

3.1 Subspaces of V

Subspaces of V of dimension r will be represented by spanning orthonormal r-frames. For each $1 \le j \le r$, let $\alpha_j = (\alpha_{1j}, \ldots, \alpha_{Mj})^T$ represent a vector $v_j \in V$, and let α be the $(M \times r)$-matrix whose entries are α_{ij}; that is, the columns of α are α_j. From Eqn. 3, it follows that $\{v_j, 1 \le j \le r\}$ is orthonormal if and only if

$$\alpha^T K \alpha = I_r\,, \tag{5}$$

where I_r is the $r \times r$ identity matrix. The collection of all $M \times r$ matrices satisfying Eqn. 5 will be denoted $\mathcal{A}$. Given $\alpha \in \mathcal{A}$, let $[\alpha]$ be the r-dimensional subspace of V associated with α; i.e.,

$$[\alpha] = \text{span}\,\{v_j, 1 \le j \le r\}\,,$$

with $v_j = \sum_{i=1}^{M} \alpha_{ij} \Phi(x_i)$. We denote by $\pi_\alpha \colon \mathbb{R}^m \to [\alpha]$ be the orthogonal projection of $\mathbb{H}$ onto $[\alpha]$. For $x \in \mathbb{R}^n$, we derive an expression for $\Phi^\alpha(x) = \pi_\alpha(\Phi(x))$. The product $\langle \Phi^\alpha(x), v_j \rangle = \langle \pi_\alpha(\Phi(x)), v_j \rangle = \langle \Phi(x), v_j \rangle$, for $1 \leq j \leq r$, can be calculated as

$$\langle \Phi(x), v_j \rangle = \sum_{i=1}^{M} \alpha_{ij} \langle \Phi(x), \Phi(x_i) \rangle = \sum_{i=1}^{M} \alpha_{ij} k(x, x_i),$$

which implies that

$$\Phi^\alpha(x) = \sum_{j=1}^{r} \left(\sum_{i=1}^{M} \alpha_{ij} k(x, x_i) \right) v_j . \tag{6}$$

3.2 Distance in $[\alpha]$

In applications using nearest-neighbor classifiers, we will be primarily interested in the distance between $\Phi^\alpha(x)$ and $\Phi^\alpha(y)$, for $x, y \in \mathbb{R}^n$. Since

$$\begin{aligned} \|\Phi^\alpha(x) - \Phi^\alpha(y)\|^2 = \langle \Phi^\alpha(x), \Phi^\alpha(x) \rangle - 2 \langle \Phi^\alpha(x), \Phi^\alpha(y) \rangle \\ + \langle \Phi^\alpha(y), \Phi^\alpha(y) \rangle , \end{aligned} \tag{7}$$

it suffices to derive expressions for inner products of the form $\langle \Phi^\alpha(w), \Phi^\alpha(z) \rangle$, $w, z \in \mathbb{R}^n$. From Eqn. 6, we obtain

$$\begin{aligned} \Phi^\alpha(w) \cdot \Phi^\alpha(z) &= \sum_{\ell=1}^{r} \left(\sum_{i=1}^{M} \alpha_{i\ell} k(w, x_i) \right) \left(\sum_{j=1}^{M} \alpha_{j\ell} k(z, x_j) \right) \\ &= \alpha^T h(w) \cdot \alpha^T h(z), \end{aligned}$$

where $h(w)$ denotes the vector $(k(w, x_1), \ldots, k(w, x_M))^T \in \mathbb{R}^{M \times 1}$, $h(z)$ is defined similarly, and $\cdot$ is the standard inner product in $\mathbb{R}^r$. In (7), we obtain

$$\begin{aligned} \|\Phi^\alpha(x) - \Phi^\alpha(y)\|^2 = \|\alpha^T h(x)\|^2 - 2\alpha^T h(x) \cdot \alpha^T h(y) \\ + \|\alpha^T h(y)\|^2. \end{aligned} \tag{8}$$

This expresses the distance solely in terms of α and the kernel function k, as desired.

4 Kernel OCA

The results of Sec. 3.1 allow us to define a performance function G for KOCA similar to the function F given by (2) . If $\alpha \in \mathcal{A}$, the definition of $G([\alpha])$ is identical to that of the function $F(U)$ in (2), with distances $d(y_{c,i}, x_{d,j}; U)$ between training and cross-validation points replaced by

$$d(y_{c,i}, x_{d,j}; [\alpha]) = \left(\|\alpha^T h(x)\|^2 - 2\alpha^T h(x) \cdot \alpha^T h(y) + \|\alpha^T h(y)\|^2 \right)^{1/2} . \tag{9}$$

To complete the description of kernel OCA, we address the problem of maximizing G over the Grassmann manifold $\mathcal{G}(V, r)$ formed by all r-dimensional subspaces of V. If $v_1, \ldots, v_m$ is an orthonormal basis of V and $e_1, \ldots, e_m$ is the standard basis of $\mathbb{R}^m$, the correspondence $e_j \mapsto v_j$ induces an identification of $\mathcal{G}(m, r)$ with $\mathcal{G}(V, r)$. This will allow us to reduce the question to an optimization problem over $\mathcal{G}(m, r)$.

4.1 Grassmann and Stiefel Manifolds

Let $\mathcal{V}(m,r)$ denote the *Stiefel* manifold of orthonormal r-frames $(u_1,\ldots,u_r)$ in $\mathbb{R}^m$, which we represent by the $m\times r$ matrix U whose jth column is u_j, $1\le j\le r$. A matrix $U\in\mathbb{R}^{m\times r}$ represents an element of $\mathcal{V}(m,r)$ if and only if $U^TU=I_r$. The Grassmann manifold $\mathcal{G}(m,r)$ may be viewed as the quotient space of $\mathcal{V}(m,r)$ under the following equivalence relation: $U_1\sim U_2$ if there exists an orthogonal matrix $H\in O(r)$ such that $U_1=U_2H$. This just formalizes the simple fact that if we represent r-planes in $\mathbb{R}^m$ using their orthonormal basis, we have to account for all possible choices. We abuse notation and use $U\in\mathbb{R}^{m\times r}$ to denote both an element in the Stiefel manifold and its equivalence class in the Grassmannian.

Let J be the $m\times r$ matrix formed by the first r columns of the identity matrix I_m. The matrix J represents the orthonormal r-frame formed by the first r elements of the standard basis of $\mathbb{R}^m$. It can be shown that tangent vectors to $\mathcal{G}(m,r)$ at J can be identified uniquely with matrices of the form

$$\begin{bmatrix} 0 & B \\ -B^T & 0 \end{bmatrix} J \in \mathbb{R}^{m\times r},$$

where $B\in\mathbb{R}^{r\times(m-r)}$. Let E_{ij}, $1\le i\le r$ and $r<j\le m$, be the $m\times m$ matrix whose (k,l) entry is

$$E_{ij}(k,l)=\begin{cases} 1/\sqrt{2}, & \text{if } k=i \text{ and } l=j;\\ -1/\sqrt{2}, & \text{if } k=j \text{ and } l=i;\\ 0, & \text{otherwise,}\end{cases}$$

Then, $\{E_{ij}J,\ 1\le i\le r,\ r<j\le m\}$ represents an orthonormal basis of the tangent space $T_J\mathcal{G}(m,r)$.

Any two orthonormal r-frames in $\mathbb{R}^m$ differ by the action of an orthogonal matrix. Thus, for any $U\in\mathcal{G}(m,r)$, there is an orthogonal matrix $Q\in O(n)$ such that $J=QU$. Any such matrix Q has the property that $Q^T=[U,W]$, where W is some $m\times(m-r)$ matrix satisfying $W^TW=I_{m-r}$ and $U^TW=0$; the role of V is to complete U to an $m\times m$ orthogonal matrix. Since left multiplication by Q^T induces an isometry on $\mathcal{G}(m,d)$, it follows that $\{Q^TE_{ij}J,\ 1\le i\le r,\ r<j\le m\}$, represents an orthonormal basis of the tangent space $T_U\mathcal{G}(m,r)$. Another important consequence of the fact that left multiplication by Q^T is an isometry is that the geodesic $\gamma_{ij}(t;U)$ in $\mathcal{G}(m,r)$ starting at U with initial velocity $Q^TE_{ij}J\in T_U\mathcal{G}(m,r)$ is given by the action of Q^T on the geodesic $\gamma_{ij}(t;J)=e^{tE_{ij}}J$. In other words,

$$\gamma_{ij}(t;U)=Q^Te^{tE_{ij}}J. \tag{10}$$

4.2 Maximizing G

Let $U\in\mathcal{G}(m,r)$. The mapping $U\mapsto[DU]$, where D is the $M\times m$ matrix whose column vectors are given by (4) and $[DU]$ denotes the subspace of V associated with $\alpha=DU$, induces an identification $\mathcal{G}(m,r)\approx\mathcal{G}(V,r)$. Thus, maximizing the performance function $G\colon\mathcal{G}(V,r)\to\mathbb{R}$ is equivalent to maximizing $H\colon\mathcal{G}(m,r)\to\mathbb{R}$, where

$$H(U)=G([DU]). \tag{11}$$

The Gradient of H. The partial derivatives of H at $U \in \mathcal{G}(m,r)$ in the direction $Q^T E_{ij} J$, $1 \leq i \leq r$, $r < j \leq m$, can be evaluated as

$$\partial_{ij} H(U) = \lim_{\epsilon \to 0} \frac{G\left([DQ^T e^{\epsilon E_{ij}} J] - G\left([DU]\right)\right)}{\epsilon}. \tag{12}$$

Note that D is fixed and Q depends only on U. To estimate the partial derivatives at U using finite differences, we first compute DQ^T. If we write the $m \times m$ identity matrix as $I_m = [e_1 \ \ldots \ e_m]$, the exponential $e^{\epsilon E_{ij}}$ can be obtained from I_m by the following column replacements:

$$e_i \mapsto \cos(\epsilon/\sqrt{2}) e_i - \sin(\epsilon/\sqrt{2}) e_j \quad \text{and} \quad e_j \mapsto \sin(\epsilon/\sqrt{2}) e_i + \cos(\epsilon/\sqrt{2}) e_j.$$

Then, the $M \times r$ matrix $DQ^T e^{\epsilon E_{ij}} J$ can be calculated by first performing the same column replacements on DQ^T and then deleting the last $m - r$ columns.

Also, to evaluate the performance function G at $[DQ^T e^{\epsilon E_{ij}} J]$, we need to compute the distances $d(y_{c,i}, x_{d,j}; [DQ^T e^{\epsilon E_{ij}} J])$ between training and cross-validation points. Since $DQ^T e^{\epsilon E_{ij}} J$ and DQ^T differ in a single column, Eqns. 9 and 7 show that significant gains in computational efficiency can be realized by first calculating $d(y_{c,i}, x_{d,j}; [DU])$ and storing the intermediate results.

To summarize, given U_t at time t, we first compute Q such that $Q^T = [U_t \, , \, W_t]$, where the columns of W_t form an orthonormal basis for the null space of U_t. Using Eqn. 12, we estimate the partial derivatives $\partial_{ij} H(U_t)$ and then add a stochastic component to $\partial_{ij} H(U_t)$ to carry out a stochastic gradient optimization of H; details of the implementation of the stochastic search can be found in [7].

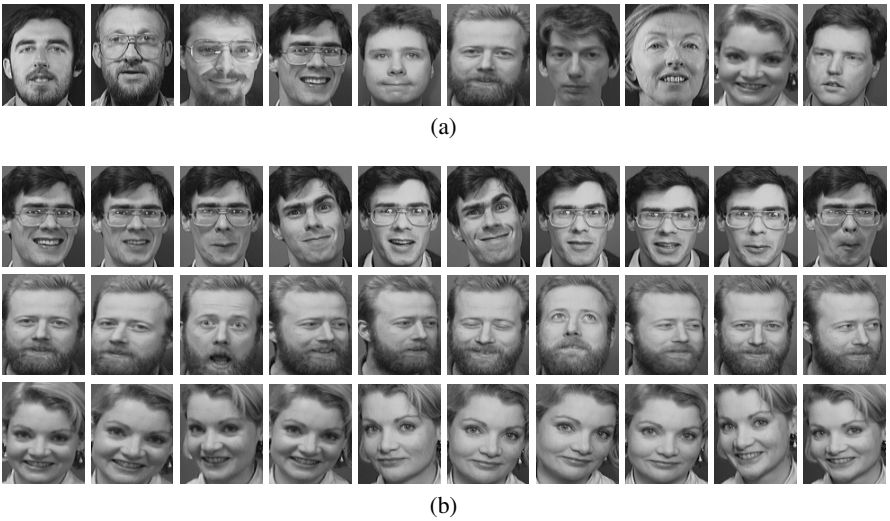


Fig. 1. Part of the ORL dataset: (a) 10 subjects used in the experiments; (b) images of three selected subjects taken at different facial expression and illumination conditions

5 Experimental Results

We present the results of several image-based object recognition experiments. By precomputing $\Phi(w, x_i), i = 1, \ldots, M$, for each validation image w, the implementation of the proposed algorithm is at least as efficient as the OCA algorithm in Euclidean space. Recall that if n is much larger than the size of the training set M, a substantial additional computational gain is realized by considering only subspaces in the span of $\Phi(x_i)$, $1 \leq i \leq M$, as remarked in Sect. 3. Compared to a direct implementation in kernel space [8], on a face recognition data set, the computational time is reduced from several hours to just a few seconds. Furthermore, the techniques developed provide an adaptive way of balancing efficiency and accuracy as illustrated in Fig. 5.

As in other gradient-based methods and the original OCA algorithms, the choice of free parameters may affect results significantly. Additionally, for KOCA, the choice of kernel functions is also important. Instead of pursuing asymptotic convergence results, we have conducted numerical simulations to demonstrate the effectiveness of the proposed algorithm. We varied the subspace dimension, as well as the kernel functions.

Using part of the ORL face database, we have applied the proposed algorithm to the search for optimal linear basis in the context of face recognition in the kernel space. The dataset consists of faces of 40 different subjects with 10 images each. The subjects are shown in Fig. 1(a) and the images of three particular subjects are shown in Fig. 1(b) to illustrate the variation of facial expression and lighting condition. Here we used 10 subjects for the plots in the Figures 2-8 and 20 subjects for the results shown in Tab. 1. Figure 2 shows the evolution of the optimization performance using a Gaussian kernel with a fixed width σ. Fig. 2(a) and (b) show two cases with random initial subspace while Fig. 2(c) shows the case using the kernel PCA as the initial subspace. In each case, the plot on the left shows the evolution of the performance function $F(U_t)$ with $\beta = 0.5$; the middle plot shows the corresponding recognition rate. Note that the recognition rate is piecewise constant and does not have a meaningful gradient for stochastic optimization while $F(U_t)$ is smooth. The right plot shows the distance of U_t from the initial one (Frobenius norm), indicating that the optimization process is effective. In all these cases, the optimization is successful in maximizing the recognition performance.

We have also used polynomial kernel of different degrees. Fig. 3 shows three such examples. As in the previous example, the performance improves significantly with the number of iterations in all the cases. Compared to the results in Fig. 2, here the performance function itself is worse than that using the Gaussian kernel, indicating the importance of the kernel function for performance.

For dimension reduction, the choice of the subspace dimension is an important parameter. Using the proposed method, we can significantly reduce the required dimension for a given level of performance. To show this, Fig. 4 shows three examples of Gaussian kernel for different values of the subspace dimension r. As expected, when r is larger, it takes fewer iterations to achieve a given performance. With $r = 3$, the proposed algorithm achieves maximum recognition performance. For applications where the computational complexity is critical, the proposed method may reduce the required dimension effectively.

As pointed out earlier, significant computational efficiency can be realized by restricting subspaces to those contained in the span of the kernelized data. To illustrate

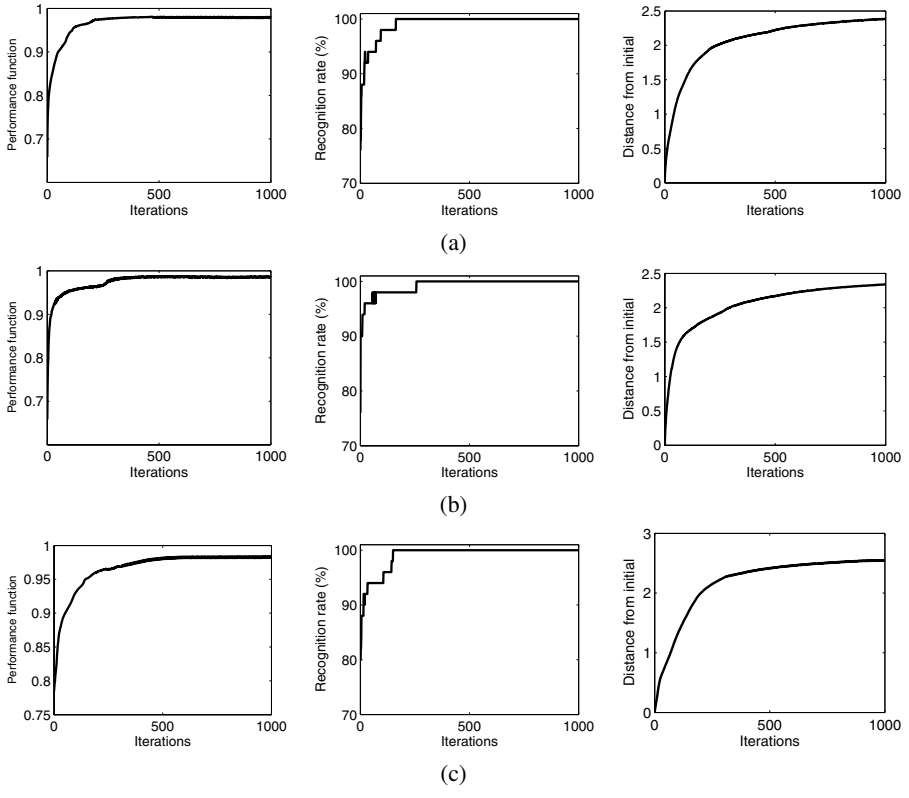


Fig. 2. Plots of the performance function F (left), the corresponding recognition rate (middle), and the distance from the initial subspace (right) versus the number t of iterations using projections onto a 4-dimensional subspace. Here Gaussian kernel with proper σ is used. (a) and (b) Two random initial subspaces. (c) Initial subspace given by KPCA, whose recognition performance is 80%.

that the gain in efficiency usually does not lead to significant loss in discriminative power, we show in Fig. 5(a) a plot of the distribution of eigenvalues of K matrix given by a Gaussian kernel; Fig. 5(b) shows the percentage of energy captured by the first given number of eigenvectors. Clearly we can reduce the dimension to a much smaller number and still have most of the information for classification. Fig. 6 shows three examples with a different number of eigenvectors. As the examples in Fig. 6(b) and (c) show, one can reduce the dimension of the search space without much loss of performance as compared to that given in Fig. 2, where the span of all the training images is used. It is expected that when the search space is reduced too much, the performance loss can become significant, as shown in Fig. 6(a). In the extreme case, when $m = r$, KOCA is reduced to KPCA or other method, depending how the initial subspace is generated.

To summarize the experiments and compare the performance using the proposed algorithm and that of KPCA [10], Tab. 1 shows the performance of both methods using

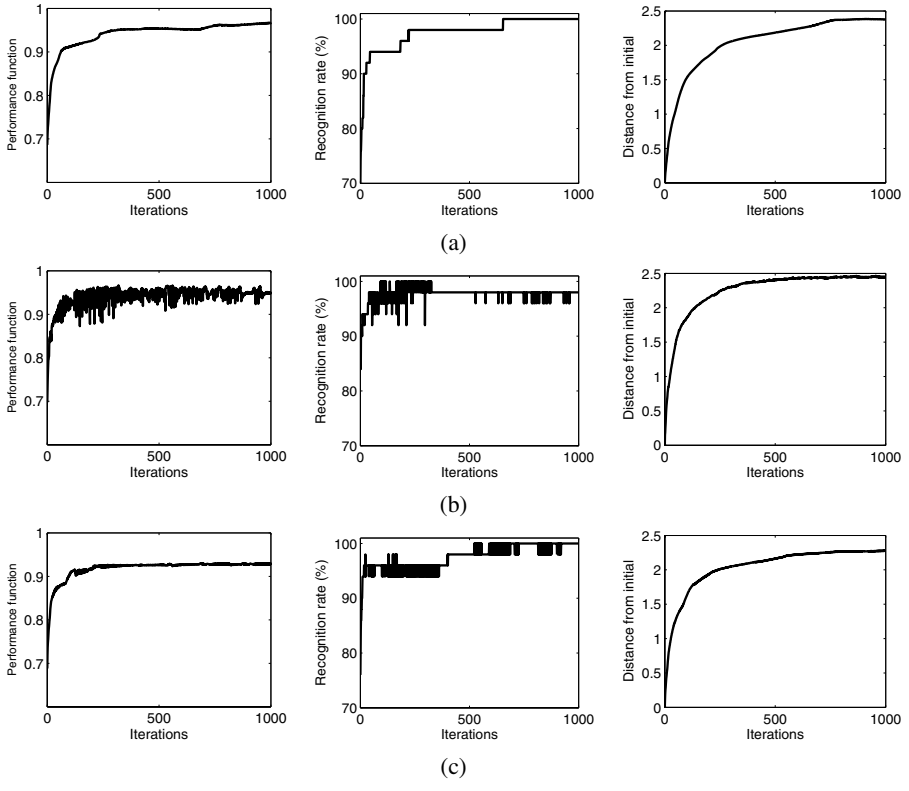


Fig. 3. Plots of the performance function F (left), the corresponding recognition rate (middle), and the distance from the initial subspace (right) versus the number t of iterations using projections onto a 4-dimensional subspace. Here polynomial kernels $k(x, y) = (x \cdot y)^d$ with different d's are used. The initial subspace is given randomly. (a) Polynomial kernel of $d = 2$. (b) Polynomial kernel of $d = 3$. (c) Polynomial kernel of $d = 4$.

different kernel functions with different dimension of subspaces (r) using 20 subjects of the ORL face dataset. It is clear that the proposed algorithm is significantly more effective than KPCA in all the cases. Additionally, this shows again the importance of kernel functions and how to learn the kernel functions is an important problem.

While the above experiments demonstrate clearly the effectiveness of the proposed KOCA technique, for real world applications, one is interested in the generalization performance, i.e., the performance on images that are not part of the training. To simulate this situation, we divide the face dataset into a training set, a cross validation set, and a separate test set, i.e., images in the test set are not used in the optimization performance. To visualize the effectiveness of KOCA, we use five classes here and set $r = 2$. Fig. 7 shows the 2-dimensional representation of the training, cross validation, and test images given by KPCA and KOCA. Here each image is shown at the center given by its 2-dimensional representation. It is clear that some of images from the same class do not form good clusters in the space given by KPCA. In comparison, images

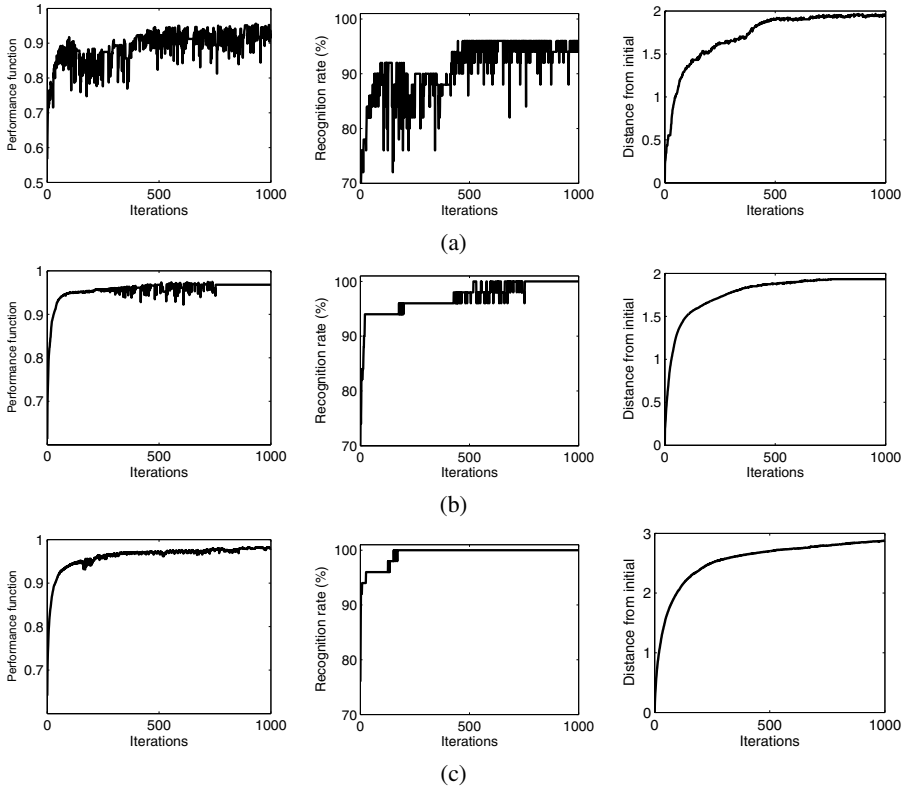


Fig. 4. Plots of the performance function F (left), the corresponding recognition rate (middle), and the distance from the initial subspace (right) versus the number t of iterations using projections onto r-dimensional subspaces with different r's. Here Gaussian kernel with proper σ is used and the initial subspace is given randomly. (a) $r = 2$. (b) $r = 3$. (c) $r = 6$.

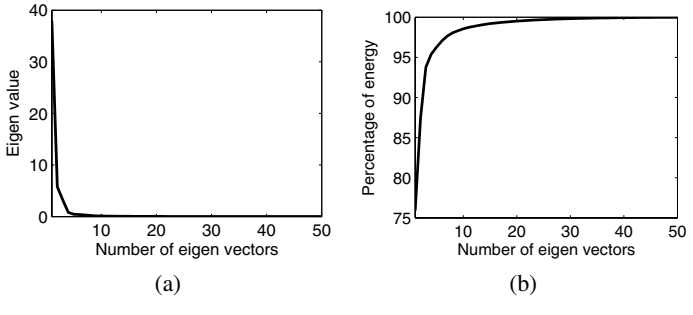


Fig. 5. Distribution of eigen values (a) and the energy captured by the first given eigen vectors (b)

from each of the five classes form a compact cluster that is away from clusters of other classes. Here we used a modified version of (2), which is related to a 4-nearest neighbor performance.

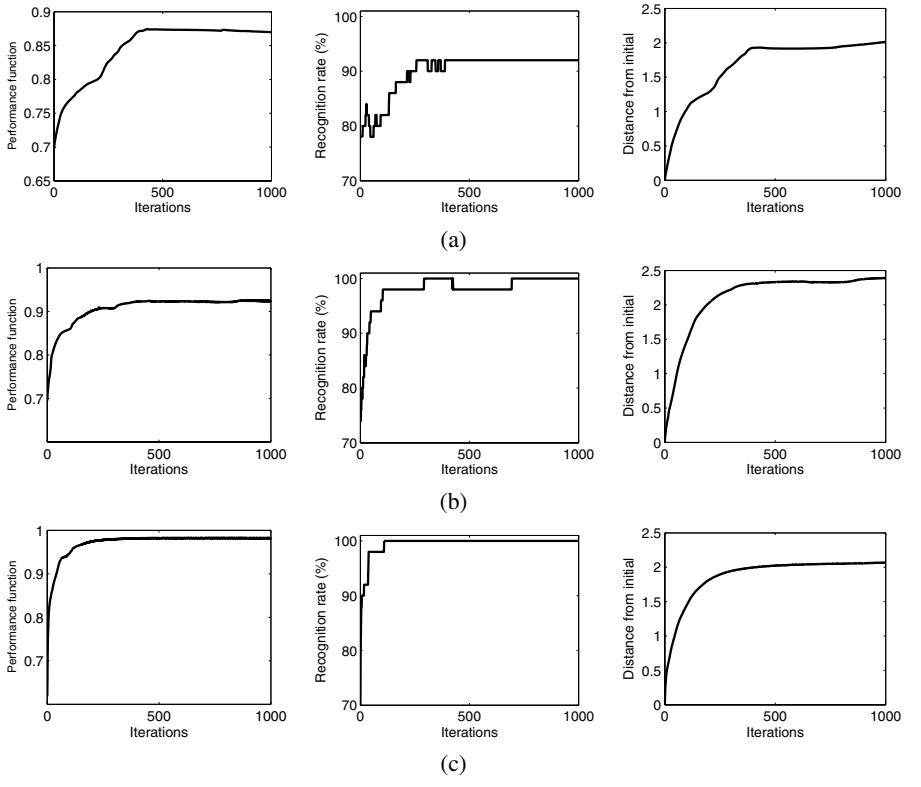
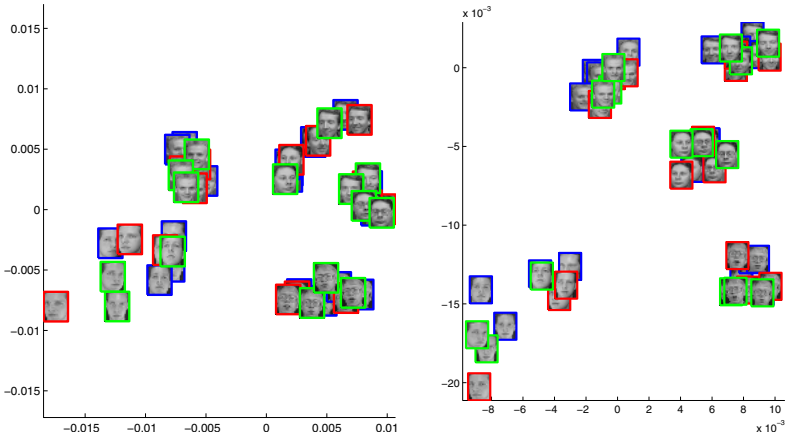


Fig. 6. Plots of the performance function F (left), the corresponding recognition rate (middle), and the distance from the initial subspace (right) versus the number t of iterations using projections onto an 4-dimensional subspace. Here Gaussian kernel with proper σ is used and the initial subspace is given randomly. (a) $m = 8$ that captures 98% of the energy. (b) $m = 16$ that captures 99.25% of the energy. (c) $m = 35$ that captures 99.99% of the energy.

Table 1. Comparison of recognition performance of KPCA and the proposed algorithm

Kernel function	Dimension r	KPCA	Proposed	Kernel function	Dimension r	KPCA	Proposed
Gaussian	2	48%	91%	$(x, y)^2$	4	82%	96%
Gaussian	4	82%	99%	$(x, y)^3$	4	80%	97%
Gaussian	6	83%	100%	$(x, y)^4$	4	82%	95%

To show the significance of KOCA, we computed the nearest neighbor classifier, the 4-nearest neighbor classifier, and $F(U)$ (given by (2)) in the original image space (each image is $92 \times 112 = 10,304$), the 2-dimensional KPCA space, and a 2-dimensional KOCA space using a Gaussian kernel. Tab. 2 shows the results. Note that while the nearest neighbor performance is high in all the cases for this small set, the 4-nearest neighbor performance is significantly different. KOCA achieves a much better 4-nearest neighbor performance due to the much better clustering structure as shown in Fig. 7(b).



(a) KPCA (b) KOCA

Fig. 7. 2-dimensional representation of training (blue), cross validation (green), and test (red) images using (a) KPCA and (b) KOCA. It is clear that clusters are much better organized using the representation given by KOCA. Here the axes are the projections given the corresponding 2-dimensional projection matrix.

Table 2. Recognition performance of different representations on a five-class subset

Set	F	Nearest neighbor (%)	4-nearest neighbor(%)
10,304-dimensional original feature			
Cross validation	0.542	100.0	86.7
Test	0.545	100.0	86.7
2-dimension KPCA feature			
Cross validation	0.719	100.0	80.0
Test	0.715	100.0	80.0
2-dimension KOCA feature			
Cross validation	0.994	100.0	100.0
Test	0.955	**100.0**	**100.0**

To demonstrate the significance of the proposed technique, we repeated the above experiments on the full ORL dataset using $r = 10$. Tab. 3 shows the performance. Clearly KOCA not only reduces the dimension of the images significantly, but also increases the performance on both the cross validation set and more importantly on the test set.

Note that the proposed technique is not limited to images and applies to any recognition problem, where the input can be represented as a vector of a fixed length. As an example, we have applied our technique on an optical character recognition (OCR) dataset from the UCI machine learning repository [1]. Since there is no cross validation set in the given setting, (2) was modified to relate to the leave-one-out performance

[1] Obtained from http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Table 3. Recognition performance of different representations on the full 40-class ORL dataset

Set	F	Nearest neighbor (%)	4-nearest neighbor(%)
10,304-dimensional original feature			
Cross validation	0.521	96.7	71.7
Test	0.524	94.2	73.3
10-dimension KPCA feature			
Cross validation	0.529	83.3	52.9
Test	0.526	92.5	63.3
10-dimension KOCA feature			
Cross validation	0.995	100.0	100.0
Test	0.937	**100.0**	**99.17**

Table 4. Recognition performance of different representations on an OCR dataset

Set	F	Nearest neighbor (%)	11-nearest neighbor(%)
Original 64-dimensional feature			
Training (leave-one-out)	0.637	95.2	94.4
Test	0.638	94.7	93.9
10-dimensional KICA feature			
Training (leave-one-out)	0.280	22.9	28.0
Test	0.288	24.4	17.8
10-dimensional KOCA feature			
Training (leave-one-out)	0.919	98.4	98.0
Test	0.862	**96.1**	**96.1**

on the training set. Tab. 4 shows the performance in the original space, the initial 10-dimensional space given by the FastICA algorithm [6] in the kernel space, and a 10-dimensional space given by KOCA that uses KICA as the initial condition. As in the previous example, KOCA not only reduces the dimensionality significantly but also improves the performance on the test compared to that in the original space.

6 Conclusion and Discussion

In this paper, we presented a kernel analogue of Optimal Component Analysis (OCA), addressing both theoretical and computational aspects of the problem. The kernel approach allows one to model nonlinearity in data structure, overcoming a fundamental limitation of OCA, as proposed in [7]. To achieve computational efficiency, the algorithms developed exploit the geometric structure of Grassmann manifolds. Several experiments were carried out and results compared to those obtained via kernel PCA.

As with other kernel methods, performance is often tied to the choice of the kernel function. Thus, in applications, the choice of the kernel function for a specific classification problem is of critical importance. To illustrate this point, Fig. 8 shows plots of the performance functions associated with three Gaussian kernels of different widths. In each column, the top panel shows a contour plot of the matrix K and the bottom panel shows the performance function with respect to the number t of iterations. Clearly the

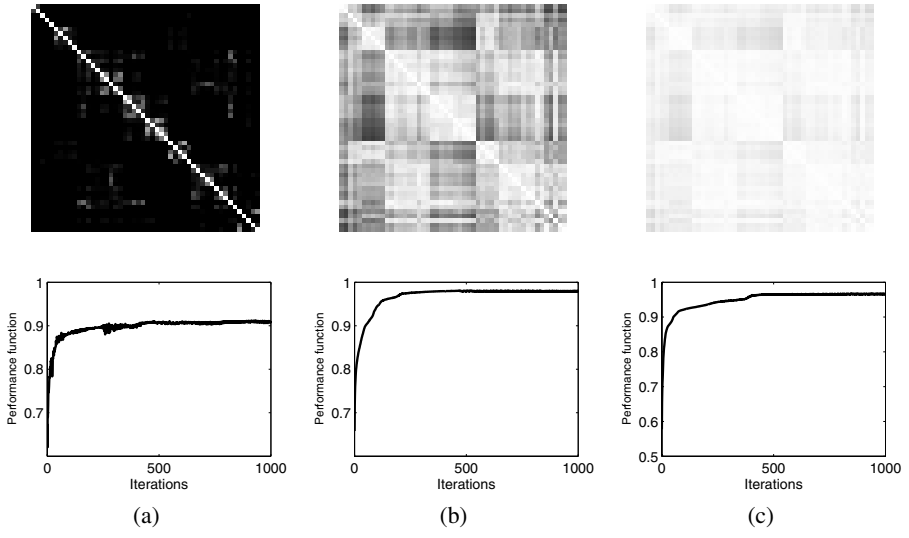


Fig. 8. The K matrix using Gaussian kernel of different σ's. subspace is given randomly. In each panel, the top image shows the K matrix and the bottom plot shows the corresponding performance function with respect to t. (a) A Gaussian kernel with σ that is too small. (b) A Gaussian kernel with a proper σ. (c) A Gaussian kernel with σ that is too large.

performance is affected by the choice of kernel function parameters. Note that in the proposed formulation one can treat the kernel function parameters in the search space and one can perform optimization in the joint space to obtain optimal subspace and kernel function parameters. This needs to be investigated further.

The geometric optimization techniques developed in this paper were applied to a performance function derived from the nearest-neighbor classifier, but they are adaptable to performance functions based on other criteria. For example, if the choice of bases is relevant in addition to the choice of subspaces, the solution space becomes a Stiefel manifold; one such criterion is to impose both sparseness and recognition performance [11]. The formulation given here can be used directly to extend the corresponding algorithms and techniques to the kernel space. Thus, the methodology developed yields a general framework and efficient algorithms for learning optimal low-dimensional representations in the presence of nonlinearity.

Acknowledgment. This work was supported in part by NSF grants IIS-0307998 and CCF-0514743, and an ARO grant W911NF-04-01-0268.

References

1. F. Bach and M. I. Jordan. Kernel independent component analysis. *Journal of Machine Learning Research*, vol. 3, pp. 1–48, 2003.
2. G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12:2385–2404, 2000.

3. P. N. Belhumeur, J. P. Hepanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
4. B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proc. of the 5th Annual Workshop on Computational Learning Theory*, pages 144–152, 1992.
5. W. M. Boothby. *An Introduction to Differential Manifolds and Riemannian Geometry*. Academic Press, 1986.
6. A. Hyvarinen. Fast and robust fixed-point algorithm for independent component analysis. *IEEE Transactions on Neural Networks*, 10:626–634, 1999.
7. X. Liu, A. Srivastava, and Kyle Gallivan, "Optimal linear representations of images for object recognition," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 26, no. 5, pp. 662–666, 2004.
8. W. Mio, Q. Zhang and X. Liu, "Nonlinearity and optimal component analysis," In *the Proceedings of the International Conference on Neural Networks*, 2005.
9. T. Poggio. On optimal nonlinear associative recall. *Biological Cybernetics*, 19:201–209, 1975.
10. B. Schölkopf, A. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
11. A. Srivastava and X. Liu, "Tools for Application-Driven Dimension Reduction," *Neurocomputing*, in press, 2005.
12. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.
13. F. W. Warner, *Foundations of Differentiable Manifolds and Lie Groups*, Springer, New York, 1983.
14. Q. Zhang and X. Liu, "Kernel optimal component analysis," In *the Proceedings of the IEEE Workshop on Learning in Computer Vision and Pattern Recognition*, 2004.