Optimal Linear Representations of Images for Object Recognition

Xiuwen Liu, *Member*, *IEEE*, Anuj Srivastava, and Kyle Gallivan

Abstract—Although linear representations are frequently used in image analysis, their performances are seldom optimal in specific applications. This paper proposes a stochastic gradient algorithm for finding optimal linear representations of images for use in appearance-based object recognition. Using the nearest neighbor classifier, a recognition performance function is specified and linear representations that maximize this performance are sought. For solving this optimization problem on a Grassmann manifold, a stochastic gradient algorithm utilizing intrinsic flows is introduced. Several experimental results are presented to demonstrate this algorithm.

Index Terms—Optimal subspaces, Grassmann manifold, object recognition, linear representations, dimension reduction, optimal component analysis.

1 Introduction

The task of recognizing objects from their 2D images generally requires excessive memory storage and computation as images are rather high-dimensional. High dimensionality also prohibits effective use of statistical techniques in image analysis since statistical models on high-dimensional spaces are difficult both to derive and to analyze. On the other hand, it is well understood that images are generated via physical processes that in turn are governed by a limited number of physical parameters. This motivates a search for methods that can reduce image dimensions without a severe loss in information. A commonly used technique is to project images linearly to some predefined low-dimensional subspaces and use these projections for image analysis. For instance, let U be an $n \times d$ orthogonal matrix denoting a basis of an orthonormal d-dimensional subspace of $I\!R^n$ ($n >> d$) and let I be an image reshaped into an $n \times 1$ vector. The vector $a(I) = U^T I \in I\!R^d$, also called the vector of image coefficients, provides a d-dimensional representation of I. Statistical methods for computer vision tasks, such as image classification, object recognition, and texture synthesis, can now be developed by imposing probability models on a.

Within the framework of linear representations, several standard bases, including principal components (PCA) and Fisher discriminant basis (FDA), have been widely used. Although they satisfy certain optimality criteria, they may not necessarily be optimal for a specific application at hand (for empirical evidence, see, e.g., [2], [8]). A major goal of this paper is to present a technique for finding *linear representations of images that are optimal for specific tasks and specific data sets*. Our search for optimal linear representation, or an optimal subspace, is based on a stochastic optimization process that maximizes a prespecified performance function over all subspaces. Since the set of all subspaces (known as the Grassmann manifold) is not a vector space, the optimization process has been modified to account for its curved geometry. Although the search for orthonormal bases can be performed using constrained optimization (with Lagrange multipliers) also, the use of intrinsic geometry of Grassmannians allows for efficiency and sophistication. In particular, it allows for MCMC (Markov chain Monte Carlo) type algorithms for optimization.

The remaining paper is organized as follows: In Section 2, we set up the problem of optimizing the recognition performance over the set of subspaces and describe a stochastic gradient technique to solve it in Section 3. Experimental results are shown in Section 4. Sections 5 concludes the paper with a brief discussion.

2 Optimal Recognition Performance

We start with a mathematical formulation of the problem. Let $U \in I\!R^{n \times d}$ be an orthonormal basis of a d-dimensional subspace of $I\!R^n$, where n is the size of an image and d is the required dimension of the optimal subspace (generally $n >> d$). For an image I, considered as a column vector of size n, the vector of coefficients is given by $a(I, U) = U^T I \in I\!R^d$. To specify a recognition performance measure F for appearance-based applications, let there be C classes to be recognized from the images; each class has k_{train} training images (denoted by $I_{c,1}, \ldots, I_{c,k_{train}}$) and k_{test} test images (denoted by $I'_{c,1}, \ldots, I'_{c,k_{test}}$) to evaluate F. In order to utilize a gradient-based algorithm, F should have continuous directional derivatives. To ensure that, we define $\rho(I'_{c,i}, U)$ to be the ratio of the between-class-minimum distance and within-class minimum distance of a test image from class c indexed by i, given by

$$\rho(I'_{c,i}, U) = \frac{\min_{c' \neq c, j} d(I'_{c,i}, I_{c',j}; U)}{\min_j d(I'_{c,i}, I_{c,j}; U) + \epsilon_0},$$

where $d(I_1, I_2; U) = \|a(I_1, U) - a(I_2, U)\|$ ($\| \cdot \|$ denotes the 2-norm) and $\epsilon_0 > 0$ is a small number to avoid division by zero. Then, define F according to:

$$F(U) = \frac{1}{Ck_{test}} \sum_{c=1}^{C} \sum_{i=1}^{k_{test}} h(\rho(I'_{c,i}, U) - 1), \tag{1}$$

where $h(\cdot)$ is a monotonically increasing and bounded function. In our experiments, we have used $h(x) = 1/(1 + \exp(-2\beta x))$, where β controls the smoothness of F. Note that $I'_{c,i}$ is classified correctly according to the nearest neighbor rule under U if and only if $\rho(I'_{c,i}, U) > 1$. It follows that F is precisely the recognition performance of the nearest neighbor classifier when $\beta \to \infty$.

Under this formulation, $F(U) = F(UH)$ for any $d \times d$ orthogonal matrix H as the distance $d(I_1, I_2; U) = d(I_1, I_2; UH)$; the choice of 2-norm in $d(I_1, I_2; U)$ allows for this equality. In other words, F depends on the subspace spanned by U but not on the specific basis chosen to represent that subspace. Therefore, our search for optimal representation(s) is on the space of d-dimensional subspaces rather than on their bases.

Let $\mathcal{G}_{n,d}$ be the set of all d-dimensional subspaces of $I\!R^n$; it is called a Grassmann manifold.[1] It is a compact, connected manifold of dimension $d(n-d)$. An element of this manifold, i.e., a subspace, can be represented either by a basis (nonuniquely) or by a projection matrix (uniquely). Choosing the former, let U be an orthonormal basis in $I\!R^{n \times d}$ such that $span(U)$ is the given subspace of $I\!R^n$. Let $[U]$ denote the set of all the orthonormal bases of $span(U)$, i.e., $[U] = \{UH | H \in I\!R^{d \times d},\ H^T H = I_d\} \in \mathcal{G}_{n,d}$. The problem of finding optimal linear subspaces for recognition becomes an optimization problem: $[\hat{U}] = \mathrm{argmax}_{[U] \in \mathcal{G}_{n,d}} F([U])$. Since the set

[1] Grassmann and Stiefel manifolds have been widely used; see, e.g., [11] for subspace tracking, [7] for motion and structure estimation, and [1], [4] for neural network learning algorithms.

- X. Liu is with the Department of Computer Science, Florida State University, Tallahassee, FL 32306. E-mail: liux@cs.fsu.edu.
- A. Srivastava is with the Department of Statistics, Florida State University, Tallahassee, FL 32306. E-mail: anuj@stat.fsu.edu.
- K. Gallivan is with the School of Computational Science and Information Technology, Florida State University, Tallahassee, FL 32306. E-mail: gallivan@cs.fsu.edu.

Manuscript received 6 Nov. 2002; revised 15 June 2003; accepted 10 Oct. 2003.
Recommended for acceptance by D. Jacobs.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117738.

0162-8828/04/$20.00 © 2004 IEEE Published by the IEEE Computer Society

$\mathcal{G}_{n,d}$ is compact and F is a smooth function, the optimizer $[\hat{U}]$ is well-defined. $[\hat{U}]$ may not be unique, i.e., it may be set-valued rather than being point-valued.

3 Optimization via Simulated Annealing

We have chosen a simulated annealing process to estimate the optimal subspace $[\hat{U}]$. In particular, we adopt a Monte Carlo version of simulated annealing using acceptance/rejection at every step and the proposal distribution results from a stochastic gradient process. Gradient processes, both deterministic and stochastic, have long been used for solving nonlinear optimization problems. Since the Grassmann manifold $\mathcal{G}_{n,d}$ is a curved space, as opposed to being a (flat) vector-space, the gradient process has to account for its intrinsic geometry. Deterministic gradients such as the Newton-Raphson method on such manifolds with orthogonality constraints have been studied in [3]. We will start by describing a deterministic gradient process (of F) on $\mathcal{G}_{n,d}$ and later generalize it to a Markov chain Monte Carlo (MCMC) type simulated annealing process.

3.1 Deterministic Gradient Flow

The performance function F can be viewed as a scalar-field on $\mathcal{G}_{n,d}$. A necessary condition for $[\hat{U}]$ to be a maximum is that, for any tangent vector at $[\hat{U}]$, the directional derivative of F in the direction of that vector should be zero. To define directional derivatives on $\mathcal{G}_{n,d}$, let J be the $n \times d$ matrix made up of first d columns of the $n \times n$ identity matrix I_n; $[J]$ denotes the d-dimensional subspace spanned to the first d axes of $\mathbb{R}^n$.

1. **Derivative of F at $[J] \in \mathcal{G}_{n,d}$:** Let E_{ij} be an $n \times n$ skew-symmetric matrix such that, for $1 \leq i \leq d$ and $d < j \leq n$,

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

Consider the products $E_{ij}J$; there are $d(n-d)$ such matrices that form an orthogonal basis of the vector space tangent to $\mathcal{G}_{n,d}$ at $[J]$. That is: $T_{[J]}(\mathcal{G}_{n,d}) = \text{span}\{E_{i,j}J\}$. Notice that any tangent vector at $[J]$ is of the form: For arbitrary scalars α_{ij},

$$\sum_{i=1}^{d} \sum_{j=d+1}^{n} \alpha_{ij} E_{ij} J = \begin{bmatrix} 0_d & B \\ -B^T & 0_{n-d} \end{bmatrix} J \in \mathbb{R}^{n \times d}, \tag{3}$$

where 0_i is the $i \times i$ matrix of zeros and B is a $d \times (n-d)$ real-valued matrix. The gradient vector of F at $[J]$ is an $n \times d$ matrix given by $A([J])J$ where

$$\begin{aligned} &A([J]) = (\sum_{i=1}^{d} \sum_{j=d+1}^{n} \alpha_{ij}(J) E_{ij}) \in \mathbb{R}^{n \times n} \\ &\text{and where } \alpha_{ij}(J) = \lim_{\epsilon \downarrow 0} \left(\frac{F([e^{\epsilon E_{ij}} J]) - F([J])}{\epsilon} \right) \in \mathbb{R}. \end{aligned} \tag{4}$$

α_{ij}s are the directional derivatives of F in the directions given by E_{ij}, respectively. The matrix $A([J])$ is a skew-symmetric matrix of the form given in (3) (to the left of J) for some B, and points to the direction of maximum increase in F, among all tangential directions at $[J]$.

2. **Derivative of F at any $[U] \in \mathcal{G}_{n,d}$:** Tangent spaces and directional derivatives at any arbitrary point $[U] \in \mathcal{G}_{n,d}$ follow similarly. For a given $[U]$, let Q be an $n \times n$ orthogonal matrix such that $QU = J$. In other words, $Q^T = [UV]$ where $V \in \mathbb{R}^{n \times (n-d)}$ is any matrix such that $V^T V = I_{n-d}$ and $U^T V = 0$. Then, the tangent space at $[U]$ is given by $T_{[U]}(\mathcal{G}_{n,d}) = \{Q^T A : A \in T_{[J]}(\mathcal{G}_{n,d})\}$, and the gradient of F at $[U]$ is an $n \times d$ matrix given by $A([U])J$ where:

$$\begin{aligned} &A([U]) = Q^T(\sum_{i=1}^{d} \sum_{j=d+1}^{n} \alpha_{ij}(U) E_{ij}) \in \mathbb{R}^{n \times n} \\ &\text{and where } \alpha_{ij}(U) = \lim_{\epsilon \downarrow 0} \left(\frac{F([Q^T e^{\epsilon E_{ij}} J]) - F([U])}{\epsilon} \right) \in \mathbb{R}. \end{aligned} \tag{5}$$

The deterministic gradient flow on $\mathcal{G}_{n,d}$ is a solution of the equation:

$$\frac{dX(t)}{dt} = A(X(t))J, \quad X(0) = [U_0] \in \mathcal{G}_{n,d} \tag{6}$$

with $A(\cdot)$ as defined in (5). Let $G \subset \mathcal{G}_{n,d}$ be an open neighborhood of $[\hat{U}]$ and $X(t) \in G$ for some finite $t > 0$. It can be shown that $X(t)$ converges to a local maximum of F, but, to achieve a global maximum, we will have to add a stochastic component to $X(t)$.

Numerical Approximation of Gradient Flow: Since the gradient of F is not available analytically, it is approximated using finite differences:

$$\alpha_{ij} = \frac{F([\tilde{U}]) - F([U])}{\epsilon}, \ 1 \leq i \leq d, \text{ and } d < j \leq n, \tag{7}$$

for a small value of $\epsilon > 0$. Here, the matrix $\tilde{U} \equiv Q^T e^{\epsilon E_{ij}} J$ is an $n \times d$ matrix that differs from U in only the ith-column which is now given by $\tilde{U}_i = \cos(\epsilon) U_i + \sin(\epsilon) V_j$, where U_i, V_j are the ith and jth columns of U and V, respectively, with V defined as earlier.

For a step size $\Delta > 0$, we will denote the search process at discrete times $X(t\Delta)$ by X_t. Then, a discrete approximation of the solution of (6) is given by:

$$\begin{aligned} &X_{t+1} = Q_t^T \exp(\Delta A_t) J, \\ &\text{where } A_t = \sum_{i=1}^{d} \sum_{j=d+1}^{n} \alpha_{ij}(X_t) E_{ij} \text{ and } Q_{t+1} = \exp(-\Delta A) Q_t. \end{aligned} \tag{8}$$

In general, the expression $\exp(\Delta A_t)$ will involve exponentiating an $n \times n$ matrix, a task that is computationally very expensive. However, given that the matrix A_t takes the skew-symmetric form before J in (3), this exponentiation can be accomplished in order $O(nd^2)$ computations, using the singular value decomposition of the $d \times (n-d)$ submatrix B contained in A_t [5]. Q_t can also be updated for the next time step using an $O(nd^2)$ update.

3.2 Simulated Annealing Using Stochastic Gradients

The gradient process $X(t)$ has the drawback that it converges only to a local maximum. For global optimization or to compute statistics under a given density on $\mathcal{G}_{n,d}$, a stochastic component is often added to the gradient process to form a stochastic gradient flow, also referred to as a diffusion process. We begin by constructing a stochastic gradient process on $\mathcal{G}_{n,d}$ and then add a Metropolis-Hastings type acceptance-rejection step to it to generate an appropriate Markov chain.

One can obtain random gradients by adding a stochastic component to (6) according to

$$dX(t) = A(X(t))Jdt + \sqrt{2T}\left(\sum_{i=1}^{d} \sum_{j=d+1}^{n} E_{ij} J \ dW_{ij}(t) \right), \tag{9}$$

where $W_{ij}(t)$ are real-valued, independent standard Wiener processes. It can be shown that (refer to [12]), under certain conditions on F, the solution of (9), $X(t)$, is a Markov process with a unique stationary probability density given by f given by:

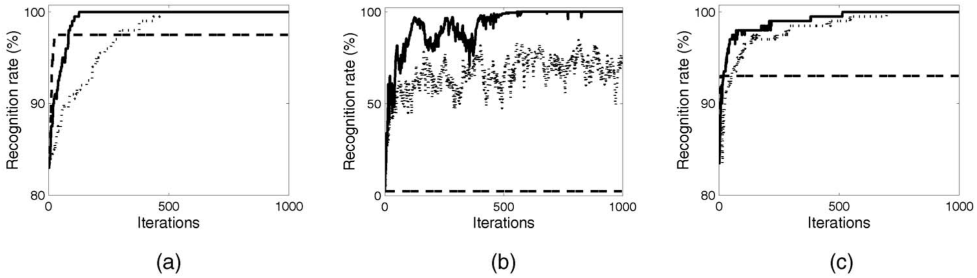

Fig. 1. Evolution of $F(X_t)$s versus t for different initial conditions. In each panel, the dashed line shows a deterministic gradient process, the dotted line a stochastic gradient process, and the solid line plots the proposed algorithm. For these curves, $n = 10,304$, (that is 92×112), $d = 5$, $k_{train} = 5$, and $k_{test} = 5$. (a) $X_0 = U_{PCA}$, (b) $X_0 = U_{ICA}$, (c) $X_0 = U_{FDA}$.

$$f([U]) = \frac{1}{Z(T)} \exp(F([U])/T), \quad \text{where } Z(T) = \int_{\mathcal{G}_{n,d}} \exp(F([U])/T). \tag{10}$$

$T \in I\!R$ denotes the temperature and f is a density with respect to the Haar measure on $\mathcal{G}_{n,d}$.

For a numerical implementation, (9) has to be discretized with some step-size $\Delta > 0$. The discretized time process is given by:

$$dA_t = A(X_t)\Delta + \sqrt{2\Delta T} \sum_{i=1}^{d} \sum_{j=d+1}^{n} w_{ij} E_{ij}, \quad X_{t+1} = Q_t^T \exp(\Delta dA_t) J, Q_{t+1} = \exp(-\Delta dA_t) Q_t, \tag{11}$$

where w_{ij}s are *i.i.d* standard normals. It can be shown that, for $\Delta \to 0$, the process $\{X_t\}$ converges to the solution of (9). $\{X_t\}$ provides a discrete implementation of the stochastic gradient process.

The stochastic component of X_t helps avoid getting stuck in a local maximum, but has the drawback of occasionally leading to states with low values of F. This drawback is removed by using an MCMC-type simulated annealing algorithm where we use the stochastic gradient process to generate a candidate for the next point but accept/reject it with appropriate probabilities. That is, the right side of (11) becomes a candidate Y that is selected as the next point X_{t+1} according to a criterion that depends on F.

Algorithm 1 MCMC Simulated Annealing: Let $X(0) = [U_0] \in \mathcal{G}_{n,d}$ be any initial condition. Set $t = 0$.

```
1. Calculate the gradient matrix $A(X_t)$ according to (5).
2. Generate $d(n-d)$ independent realizations, $w_{ij}$s, from
   standard normal density. Using the value of $X_t$, calculate
   a candidate value $Y$ according to (11).
3. Compute $F(Y)$, $F(X_t)$, and set $dF = F(Y) - F(X_t)$.
4. Set $X_{t+1} = Y$ with probability $\min\{\exp(dF/T_t), 1\}$, else set
   $X_{t+1} = X_t$. (note that $\exp(dF/T_t) = \frac{f(Y)}{f(X_t)}$).
5. Set $T_{t+1} = T_t/\gamma$, $t = t + 1$, and go to Step 1.
```

Here, $\gamma > 1$ is the cooling ratio for simulated annealing with a typical value of 1.0025. This algorithm is a particularization of A.20 [9, p. 200] where its convergence properties are discussed. The limiting point $X_* = \lim_{t \to \infty} X_t$ can be shown to be the maximizer $[\hat{U}]$.

A brute force implementation of Algorithm 1 will be computationally expensive. However, note that the discrete update rules (given in (11)) can be achieved in $O(nd^2)$ as mentioned earlier; α_{ij}s (given by (7)) can also be evaluated efficiently by exploiting the facts that 1) $\tilde{U}$ and U differ only in one column (as mentioned earlier) and 2) since ϵ is small, $\rho(I'_{c,i}, \tilde{U})$ can be approximated efficiently using $\rho(I'_{c,i}, U)$ and, thus, $F([\tilde{U}])$ can be computed efficiently also. This implementation yields an $O(nd^2)$ complexity for each iteration. One can further reduce the computational cost using a hierarchical optimization process [13]. In the context of object recognition using linear representations, we term our technique *optimal component analysis* (OCA).

4 Experimental Results

We have applied the proposed algorithm to research for optimal linear bases on a variety of data sets. Due to limited space, we presents results using only two data sets: the ORL face recognition data set[2] and the CMU PIE data set. The ORL data set consists of faces of 40 different subjects with 10 images each. The CMU PIE data set is a comprehensive face data set consisting of images of 66 subjects under a variety of conditions [10]. However, since not all images are cropped precisely, we use only those frontal images (under different lighting conditions) that are cropped manually.

4.1 Optimizing Performance Using Algorithm 1

Similar to all gradient-based methods, the choice of free parameters, such as Δ, ϵ, d, k_{train}, k_{test}, and U_0, may have a significant effect on the results of Algorithm 1. While limited theoretical results are available to analyze the convergence of such algorithms in $I\!R^n$, the case of simulated annealing over the space $\mathcal{G}_{n,d}$ is considerably more difficult. Instead of pursuing asymptotic convergence results, we have conducted extensive numerical simulations to demonstrate the convergence of the proposed algorithm, under a variety of values for the free parameters.

To support the choice of MCMC-type stochastic algorithm, Fig. 1 shows three examples (with different initial conditions) comparing 1) the deterministic gradient algorithm, 2) the stochastic gradient algorithm (acceptance probability is set to one), and 3) the proposed MCMC stochastic algorithm. Throughout this paper, U_{ICA} is computed using the FastICA algorithm by Hyvarinen [6] and U_{FDA} is calculated based on the procedure given by Belhumeur et al. [2]. While the deterministic gradient algorithm can be effective in some cases, it generally ends up in a local maximum. On the other hand, the stochastic gradient algorithm does not suffer from the problem of local maximum although its convergence is much slower than the proposed algorithm. As shown in these examples and the ones in Figs. 2 and 3, the proposed algorithm converges quickly under different kinds of conditions. We attribute this mainly to the fact that, taking the geometry of the manifold into account, the gradient flow provides the most efficient way for updating [1]. (In fact, Amari [1] has shown that such a dynamical system is Fisher efficient.) Obviously, the effectiveness of the proposed algorithm depends on the choice of parameters; through experiments we have found that it works for a wide range of parameter values.

[2] http://www.uk.research.att.com/facedatabase.html.

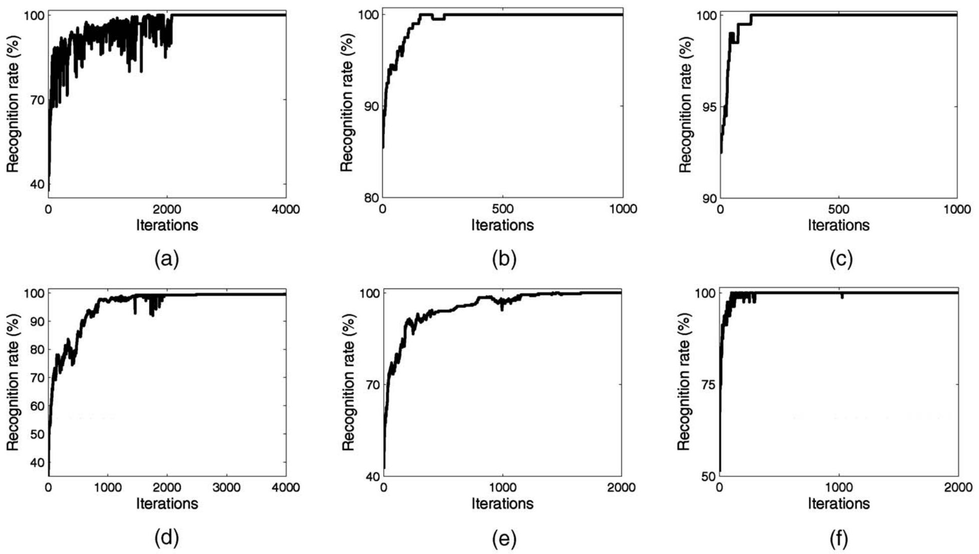

Fig. 2. Evolution of $F(X_t)$ versus t for different values of d and k_{train}. (a) $d = 3$ and $k_{train} = 5$. (b) $d = 10$ and $k_{train} = 5$. (c) $d = 20$ and $k_{train} = 5$. (d) $d = 5$ and $k_{train} = 1$. (e) $d = 5$ and $k_{train} = 2$. (f) $d = 5$ and $k_{train} = 8$.

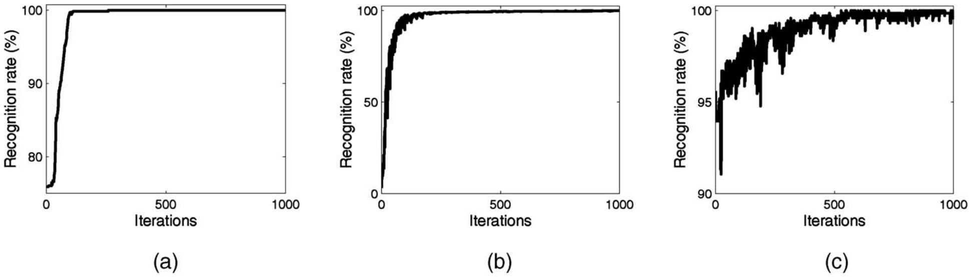

Fig. 3. Three examples of $F(X_t)$ versus t on the CMU PIE data set. Here, $n = 100 \times 100$. (a) $X_0 = U_{PCA}$ and $d = 10$. (b) $X_0 = U_{ICA}$ and $d = 10$. (c) $X_0 = U_{FDA}$ and $d = 5$.

We have studied the variation of optimal performance versus the subspace rank denoted by d and k_{train} using the ORL data set. Figs. 2a, 2b, and 2c show the results for three different values of d with $k_{train} = 5$, $k_{test} = 5$. In Fig. 2a, for $d = 3$, it takes about 2,000 iterations for the process to converge to a solution with perfect performance while, in Fig. 2c, for $d = 20$, it takes less than 200 iterations. This is expected as larger d implies a bigger space and, hence, an improved performance or easier to achieve perfect performance. Figs. 2d, 2e, and 2f show three results with three different values of k_{train} with $n = 154$ and $d = 5$. Here, the division of images into training and test sets was chosen randomly. In view of the nearest neighbor classifier being used to define F, it is easier to obtain a perfect solution with more training images. The experimental results support that observation. Fig. 2d shows the case with $k_{train} = 1$ ($k_{test} = 9$) where it takes about 3,000 iterations for the process to converge to a perfect solution. In Fig. 2f, where $k_{train} = 8$ ($k_{test} = 2$), the process converges to a perfect solution in about 300 iterations.

As another example, we have applied the proposed algorithm to a subset of the CMU PIE data set [10]. The subset we have used includes the frontal images (with lighting variations) that were cropped manually. Fig. 3 shows three examples of the algorithmic results. As in the previous examples, the proposed algorithm improves the performance significantly, often converging to solutions with perfect recognition performance.

4.2 Comparisons with Standard Subspaces

So far, we have described results on finding optimal subspaces under different conditions. In this section, we focus on comparing empirically the performances of these optimal subspaces with the frequently used subspaces, namely, U_{PCA}, U_{ICA}, and U_{FDA}.

Figs. 4a and 4b show the recognition performance (for the ORL database) with different d and k_{train} for four different kinds of subspaces:

1. optimal subspace X^* computed using Algorithm 1,
2. U_{PCA},
3. U_{ICA}, and
4. U_{FDA}.

These results highlight the fact that the recognition performance of linear representations can vary significantly depending on the parameters and data sets. To reach any conclusion, a comparison of a few standard bases is not enough and some technique such as the one proposed here seems necessary.

The above examples represent experimental situations where the performance measure F can be evaluated over the whole database. In practice, however, we often have a limited number of training images and we are interested in linear subspaces that lead to better performance on an unknown test set. To simulate this setting, we have modified ρ to be

$$\rho(I_{c,i}, U) = \frac{\min_{c' \neq c,j} d(I_{c,i}, I_{c',j}; U)}{\min_{j \neq i} d(I_{c,i}, I_{c,j}; U) + \epsilon_0}.$$

This definition is related to the leave-one-out recognition performance on the training set. We have applied this modified measure on the ORL data set by randomly dividing all the images into a nonoverlapping training and test set. Fig. 4d shows the recognition performance on a separate test set of X_t by maximizing the

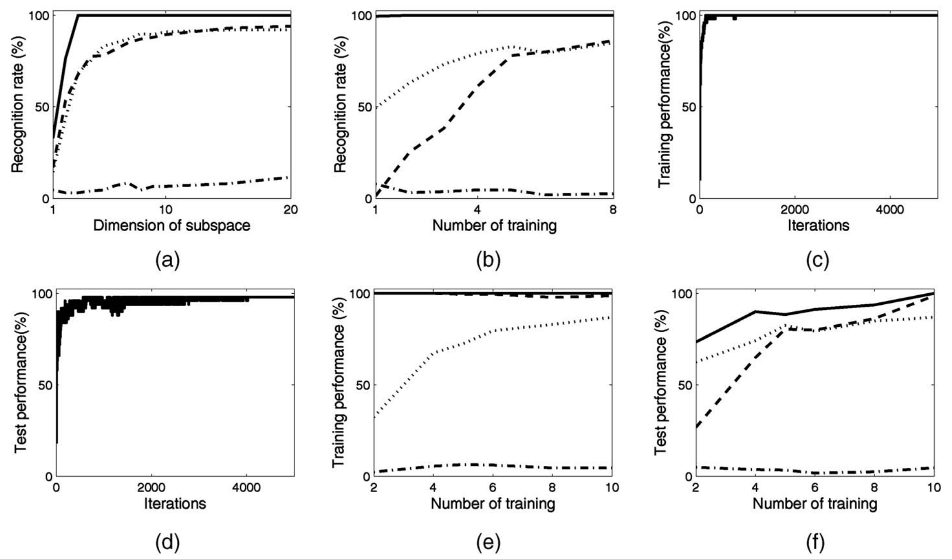

Fig. 4. (a) and (b) F of different linear subspaces versus d ($k_{train} = 5$) and k_{train} ($d = 5$), respectively, on the ORL data set. Solid line is $F(X^*)$, dashed line is $F(U_{FDA})$, dotted line is $F(U_{PCA})$, and dash-dotted line is $F(U_{ICA})$. (c) Evolution of $F(X_t)$ versus t, where $F(X_t)$ is defined using the training set only. (d) The corresponding performance on a separate test set of X_t given in (c). (e) and (f) leave-one-out recognition on the training set and a separate test set with $d = 5$ using the same legend in (a) and (b).

performance on the training set only, which is given in Fig. 4c. Fig. 4e shows the leave-one-out recognition performance on the training images and Fig. 4f the performance on a separate test set of the optimal subspaces along with common linear representations. The optimal subspaces found using only the training set also provide better performance on the test set in all these cases. Such results point to the possibility of improving generalization using the proposed algorithm, which requires further exploration.

5 Discussion

We have proposed an MCMC-based simulated annealing algorithm to find the optimal linear subspaces assuming that the performance function F can be computed. By formulating an optimization problem on the Grassmann manifold, an intrinsic optimization algorithm is presented. Extensive experiments demonstrate the effectiveness and feasibility of this algorithm.

While the focus here has been recognition, the proposed algorithm can be extended to any performance function. In fact, we have applied the algorithm and its generalized versions to 1) finding optimal linear filters that are both sparse and effective for recognition and 2) finding optimal representations for image retrieval applications. While F given by (1) is based on the nearest neighbor rule, it can be easily generalized to other classifiers such as support vector machines.

Acknowledgments

The authors thank the reviewers for their insightful comments and the producers of the ORL and PIE data sets for making them public. They also thank Eric Klassen of Florida Sate University for his help in understanding the geometry of Grassmann manifolds. This research was supported in part by US grants NSF DMS-0101429, NSF CCR-9912415, NSF IIS-0307998, ARO DAAD19-99-1-0267, and NIMA NMA 201-01-2010.

References

[1] S. Amari, “Natural Gradient Works Efficiently in Learning,” *Neural Computation,* vol. 10, pp. 251-276, 1998.

[2] P.N. Belhumeur, J.P. Hepanha, and D.J. Kriegman, “Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 7, pp. 711-720, 1997.

[3] A. Edelman, T. Arias, and S.T. Smith, “The Geometry of Algorithms with Orthogonality Constraints,” *SIAM J. Matrix Analysis and Applications,* vol. 20, no. 2, pp. 303-353, 1998.

[4] S. Fiori, “A Theory for Learning by Weight Flow on Stiefel-Grassman Manifold,” *Neural Computation,* vol. 13, pp. 1625-1647, 2001.

[5] K. Gallivan, A. Srivastava, X. Liu, and P. VanDooren, “Efficient Algorithms for Inferences on Grassmann Manifolds,” *Proc. 12th IEEE Workshop Statistical Signal Processing,* 2003.

[6] A. Hyvarinen, “Fast and Robust Fixed-Point Algorithm for Independent Component Analysis,” *IEEE Trans. Neural Networks,* vol. 10, pp. 626-634, 1999.

[7] Y. Ma, J. Kosecka, and S. Sastry, “Optimization Criteria and Geometric Algorithms for Motion and Structure Estimation,” *Int'l J. Computer Vision,* vol. 44, no. 3, pp. 219-249, 2001.

[8] A.M. Martinez and A.C. Kak, “PCA versus LDA,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 2, pp. 228-233, Feb. 2001.

[9] C.P. Robert and G. Casella, *Monte Carlo Statistical Methods.* Springer, 1999.

[10] T. Sim, S. Baker, and M. Bsat, “The CMU Pose, Illumination, and Expression Database,” *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 12, pp. 1615-1618, Dec. 2003.

[11] A. Srivastava, “A Bayesian Approach to Geometric Subspace Estimation,” *IEEE Trans. Signal Processing,* vol. 48, no. 5, pp. 1390-1400, 2000.

[12] A. Srivastava, U. Grenander, G.R. Jensen, and M.I. Miller, “Jump-Diffusion Markov Processes on Orthogonal Groups for Object Recognition,” *J. Statistical Planning and Inference,* vol. 103, nos. 1-2, pp. 15-37, 2002.

[13] Q. Zhang, X. Liu, and A. Srivastava, “Hierarchical Learning of Optimal Linear Representations,” *Proc. IEEE Workshop Statistical Analysis in Computer Vision,* 2003.