

Optimal Linear Representations of Images for Object Recognition

Xiuwen Liu

Department of Computer Science
Florida State University
Tallahassee, FL 32306

Anuj Srivastava

Department of Statistics
Florida State University
Tallahassee, FL 32306

Kyle Gallivan

School of Computational Science
and Information Technology
Florida State University
Tallahassee, FL 32306

Abstract

Simplicity of linear representations (of images) makes them a popular tool in imaging analysis applications such as object recognition and image classification. Although several linear representations, namely PCA, ICA, and FDA, have frequently been used, these representations are generally far from optimal in terms of actual application performance. We argue that representations should be chosen with respect to the application and the databases involved. Fixing an application, say object recognition, and assuming that recognition performance is computable for any linear basis (given a classifier and a database), we propose a Monte Carlo simulated annealing method that leads to optimal linear representations by maximizing the recognition performance over all fixed-rank subspaces. We illustrate this method on two popular databases.

1 Introduction

In general the task of recognizing objects from their 2D images requires excessive memory storage and computation, as images are rather high-dimensional. High dimensionality of images 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 pre-defined low-dimensional linear subspaces, and use the projections for analysis. For instance, let U be an $n \times d$ orthogonal matrix denoting a basis of an orthonormal d -dimensional subspace of \mathbb{R}^n ($n \gg d$) and let I be an image reshaped into an $n \times 1$ vector. Then, the vector $a(I) = U^T I \in \mathbb{R}^d$, also called the vector of coefficients, can be 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 bases, resulting from different optimality criteria, have been proposed. For example, principal component analysis (PCA) results in the basis, call them U_{PCA} , that are optimal in reconstructing a set of training images in the Euclidean error sense and the principal directions are those with maximal variances. Naturally, PCA may not be optimal for a recognition task since the goal in recognition is to optimally differentiate/associate images and not just to maximally capture their variance. Assuming that the underlying probability distribution (of coefficients) for each class is Gaussian, the Fisher discriminant analysis (FDA) provides an optimal linear discriminant basis U_{FDA} . However, FDA is only optimal when the underlying distributions are Gaussian and a linear discriminant function is used. It has been shown convincingly (see [11]) that images, under arbitrary linear representations, are highly non-Gaussian. Furthermore, many popular classifiers such as the nearest neighbor are nonlinear. These two factors make FDA sub-optimal in theory and in practice (as shown empirically on real datasets, see e.g. [7]). To impose statistical independence between the coefficients, independent component analysis has been proposed, leading to a linear representation we will name U_{ICA} . Independence, however, is meaningful only on a large ensemble of data, and therefore, independent components may not provide optimal linear basis for recognition.

An important and fundamental question is: How to find a linear basis that is optimal for the purpose of object recognition? Or, for any other task with a well-defined performance criterion? In general, the recognition performance depends on the choice of the training set, the test set and the classifier. Therefore, the optimal basis should also depend on those items. Linear representations based on some universal criteria will be sub-optimal when applied to arbitrary databases, as has already been observed in the literature (see, e.g. [2, 7]). 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 datasets. Our search for optimal linear representation, or an optimal subspace, is based on a stochastic optimization process that maximizes the performance function over all subspaces. Since the set of all subspaces is not a vector space, the optimization process has been modified to account for the geometry of the underlying space.

This 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. Section 5 concludes the paper.

2 Optimal Recognition Performance

We start with a mathematical formulation of the problem. Let $U \in \mathbb{R}^{n \times d}$ be an orthonormal basis of a d -dimensional subspace of \mathbb{R}^n , where n is the size of an image and d is the required dimension of the optimal subspace ($n \gg d$). For an image I , considered as a column vector of size n , the vector of coefficients is given by $a(I, U) = U^T I \in \mathbb{R}^d$. We choose to use the nearest neighbor criterion based on Euclidean metric in \mathbb{R}^d for object recognition (see Sect. 4 for justifications; other classifiers can also be used as long as the recognition performance is invariant to change of basis.). Under this implementation, the distance between two images I_1 and I_2 is defined as $d(I_1, I_2; U) = \|a(I_1, U) - a(I_2, U)\|$, where $\|\cdot\|$ denotes the 2-norm of a vector. This distance depends on the subspace spanned by U but not on the specific basis chosen to represent that subspace. That is, $d(I_1, I_2; U) = d(I_1, I_2; UO)$ for all image pairs I_1, I_2 , and for any $d \times d$ orthogonal matrix O . 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 \mathbb{R}^n ; it is called a Grassmann manifold [3]. It is a compact, connected manifold of dimension $d(n-d)$. An element of this manifold, i.e. a subspace, can be represented in several ways. Let U be an orthonormal basis in $\mathbb{R}^{n \times d}$ such that $\text{span}(U)$ is the given subspace. Then, for any $d \times d$ orthogonal matrix O , UO is also an orthonormal basis of the same subspace. There is an equivalence class of bases that span the same subspace:

$$[U] = \{UO | O \in \mathbb{R}^{d \times d}, O^T O = I_d\} \in \mathcal{G}_{n,d},$$

and throughout this paper our reference to U denotes the whole equivalence class $[U]$. Let U be such a basis of a d -dimensional subspace, and let $F(U)$ be a recognition performance measure associated with a recognition system that uses U as a linear representation. If the definition of F utilizes the metric $d(I_1, I_2; U)$ defined above, then we have that $F(U) = F(UO)$ for any $d \times d$ orthogonal matrix O ,

and therefore F is defined on the space of subspaces. That is, $F : \mathcal{G}_{n,d} \mapsto \mathbb{R}_+$ is the performance function and we want to search for the optimal subspace defined as:

$$\hat{U} = \underset{U \in \mathcal{G}_{n,d}}{\operatorname{argmax}} F(U). \quad (1)$$

Since the set $\mathcal{G}_{n,d}$ is compact and F is assumed to be a smooth function, the optimizer \hat{U} is well defined. Note that the maximizer of F may not be unique and hence \hat{U} may be set-valued rather than being point-valued. We will perform the search in a probabilistic framework by defining a probability density function

$$f(U) = \frac{1}{Z(T)} \exp(F(U)/T), \quad (2)$$

where $T \in \mathbb{R}$ plays the role of temperature and f is a density with respect to the Haar measure on the set $\mathcal{G}_{n,d}$.

3 Optimization via Simulated Annealing

We have chosen a simulated annealing process to estimate the optimal subspace \hat{U} . Gradient processes, both deterministic and stochastic, have long been used for solving non-linear optimization problems [4, 5]. 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. 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.

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. The directional derivatives on $\mathcal{G}_{n,d}$ are defined as follows. Let J be the $n \times d$ matrix formed by first d columns of $n \times n$ identity matrix, and E_{ij} be an $n \times n$ skew-symmetric matrix such that: for $1 \leq i \leq d$ and $d < j \leq n$,

$$E_{ij}(k, l) = \begin{cases} 1 & \text{if } k = i, l = j \\ -1 & \text{if } k = j, l = i \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

There are $d(n-d)$ such matrices and the collection $E_{ij}J$ forms an orthogonal basis of the vector space tangent to $\mathcal{G}_{n,d}$ at J . Notice that any linear combination of these matrices is of the following form: for arbitrary scalars α_{ij} ,

$$\sum_{i=1}^d \sum_{j=d+1}^n \alpha_{ij} E_{ij} J = \begin{bmatrix} 0_d & B \\ -B^T & 0_{n-d} \end{bmatrix} J \in \mathbb{R}^{n \times d}, \quad (4)$$

where 0_i is the $i \times i$ matrix of zeros and B is a $d \times (n - d)$ real-valued matrix. Let Q be an $n \times n$ rotation matrix such that $Q^T U = J$. The gradient vector of F at any point U is defined to be a skew-symmetric matrix given by:

$$A(U) = Q^T \left(\sum_{i=1}^d \sum_{j=d+1}^n \alpha_{ij}(U) E_{ij} \right) \in \mathbb{R}^{n \times n},$$

$$\text{where } \alpha_{ij}(U) = \lim_{\epsilon \downarrow 0} \left(\frac{(F(Q^T e^{\epsilon E_{ij}} J) - F(U))}{\epsilon} \right). \quad (5)$$

α_{ij} s are the directional derivatives of F in the directions given by E_{ij} , respectively. The matrix $A(U)$ is of the form given in Eqn. 4 for some B , and points to the direction of maximum increase in F , among all tangential directions at $U \in \mathcal{G}_{n,d}$. With this definition of the gradient vector, an updating equation is given by:

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

Let $V \subset \mathcal{G}_{n,d}$ be an open neighborhood of \hat{U} and $X(t) \in V$ for some finite $t > 0$. Define $\{U \in \mathcal{G}_{n,d} : F(U) \geq \gamma, \gamma \in \mathbb{R}_+\}$ to be the level sets of F . If the level sets of F are strictly geodesically convex in V , then the gradient process converges to a local maximum, i.e. $\lim_{t \rightarrow \infty} X(t) = \hat{U}$.

2. Simulated Annealing Using Stochastic Gradients

The gradient process $X(t)$ has the drawback that it converges only to a local maximum, which may not be useful in general. 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 diffusion [5]. Both simulated annealing and stochastic gradients [9] have frequently been used to seek global optimizers [4]. We describe an MCMC version of the simulated annealing technique that uses stochastic gradients for sampling from the proposal density.

We begin by describing stochastic gradients. One can obtain random gradients by adding a stochastic component to Eqn. 6 according to

$$dX(t) = A(X(t))dtJ + \sqrt{2T} \left(\sum_{i=1}^d \sum_{j=d+1}^n E_{ij} dW_{ij}(t) \right) J, \quad (7)$$

where $W_{ij}(t)$ are real-valued, independent standard Wiener processes. It can be shown that (refer to [10]), under certain conditions on F , the solution of Eqn. 7, $X(t)$, is a Markov process with a unique stationary probability density given by f (Eqn. 2). For a numerical implementation, Eqn. 7 has to be discretized. For a step-size $\Delta > 0$, the discrete time process is implemented using the following equations:

$$\begin{aligned} d\tilde{X}_t &= A(X_t)\Delta J + \sqrt{2\Delta T} \sum_{i=1}^d \sum_{j=d+1}^n w_{ij} E_{ij} J, \\ X_{t+1} &= Q_t^T \exp(\Delta d\tilde{X}_t) J, \\ Q_{t+1} &= \exp(-\Delta d\tilde{X}_t) Q_t, \end{aligned} \quad (8)$$

where w_{ij} 's are *i.i.d* standard normals. It is shown in [1] that for $\Delta \rightarrow 0$, the process $\{X_t\}$ converges to the solution of Eqn. 7. The process $\{X_t\}$ provides a discrete implementation of the stochastic gradient process.

In case of MCMC simulated annealing, we use this stochastic gradient process to generate a candidate for the next point along the process but accept it only with a certain probability. That is, the right side of the second equation in Eqn. 8 becomes a candidate Y that may or may not be selected as the next point X_{t+1} .

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 Eqn. 5.
2. Generate $d(n-d)$ independent realizations, w_{ij} s, from standard normal density. With X_t , calculate the candidate value Y according to Eqn. 8.
3. Compute $F(Y)$, $F(X_t)$, and set $dF = F(Y) - F(X_t)$.
4. Set $X_{t+1} = Y$ with probability $\min\{\exp(dF/T), 1\}$, else set $X_{t+1} = X_t$.
5. Modify T , set $t = t + 1$, and go to Step 1.

The resulting process X_t forms a Markov chain and let X^* be a limiting point of this Markov chain, i.e. $X^* = \lim_{t \rightarrow \infty} X_t$. This algorithm is a particularization of Algorithm A.20 (p. 200) in the book by Robert and Casella [9]. Please consult that text for the convergence properties of X_t .

4 Experimental Results

We have applied the proposed algorithm to the search for optimal linear bases in the context of object recognition. Once again we emphasize that this algorithm requires evaluation, exact or approximate, of F , the recognition performance, for any linear basis U . So far we have not specified a performance function F but we do so now to illustrate the experimental results. We choose a nearest neighbor rule, under the Euclidean metric on the coefficients, as the classifier because it is possible to estimate the performance efficiently when the bases are changed. In addition, given sufficient amount of training data, the asymptotic error under this rule is bounded to be within two times of the Bayesian error.

Definition of F : To specify F , let there be C classes to be recognized from the images; each class has k_{train} training images (denoted by $I_{c,1}, \dots, I_{c,k_{train}}$) and k_{test} test images (denoted by $I'_{c,1}, \dots, I'_{c,k_{test}}$) to evaluate the recogni-

tion performance measure. In order to utilize the stochastic gradient proposal in simulated annealing, F should be a function with continuous directional derivatives. Since the decision function of the nearest neighbor classifier is discontinuous, the resulting recognition performance function is discontinuous and needs modification. To obtain a smooth function F , 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 i from class c , given by

$$\rho(I'_{c,i}, U) = \frac{\min_{c' \neq c, j} d(I'_{c,i}, I'_{c',j}; U)}{\min_j d(I'_{c,i}, I'_{c,j}; U) + \epsilon}, \quad (9)$$

where $d(I_1, I_2; U) = \|\alpha(I_1, U) - \alpha(I_2, U)\|$ as given before, and $\epsilon > 0$ is a small number to avoid division by zero. Then, define F according to:

$$F(U, \beta) = \frac{1}{C k_{test}} \sum_{c=1}^C \sum_{i=1}^{k_{test}} h(\rho(I'_{c,i}, U) - 1, \beta), \quad (10)$$

where $h(\cdot, \cdot)$ is a monotonically increasing and bounded function in its first argument. In our experiments, we have used $h(x, \beta) = 1/(1 + \exp(-2\beta x))$, where β controls the smoothness of F . It follows that F is precisely the recognition performance of the nearest neighbor classifier when we let $\beta \rightarrow \infty$.

Description of Databases: Two types of databases have been used in our experiments: the ORL face recognition dataset¹ and the COIL dataset [8]. The ORL dataset consists of faces of 40 different subjects with 10 images each. The full COIL database consists of 7200 images at different azimuthal angles of 100 3-D objects with 72 images each. In this paper, we have used a part of the COIL database by involving only the first 20 objects, with a total of 1,440 images.

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 \mathbb{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.

As a first set of results, we run the simulated annealing algorithm with different initial conditions. Fig. 1 - 3 show

the results on the ORL database with different initial conditions and Fig. 4 shows similar examples on the COIL-20 database. These results underscore two important points about Algorithm 1: (i) the algorithm is consistently successful in seeking optimal linear basis from a variety of initial conditions, and (ii) the algorithm moves effectively on the manifold $\mathcal{G}_{n,d}$ with the final solution being far from the initial condition.

Fig. 1 shows case when X_0 is set to U_{PCA} , U_{ICA} , or U_{FDA} . In each row, the left figure plots $F(X_t)$ versus t while the right figure plots the distance between X_t and X_0 versus t . FDA was calculated using a procedure given in [2] and ICA was calculated using a FastICA algorithm proposed by Hyvärinen [6]. In these experiments, $n = 154$, $d = 5$, $k_{train} = 5$, and $k_{test} = 5$. While these commonly used linear bases provide a variety of performances, the proposed algorithm converges to a perfect classification solution regardless of the initial condition. Here the distance between any two subspaces U_1 and U_2 is computed as: $\|U_1 U_1^T - U_2 U_2^T\|$. Keep in mind that in $\mathcal{G}_{n,d}$, the maximum distance between any two d -dimensional subspaces can only be $\sqrt{2d}$. The distance plots highlight the fact that the algorithm moves effectively on the Grassmann manifold going large distances along the chains. We found multiple subspaces that lead to perfect classification and these three optimal solutions are quite different from each other.

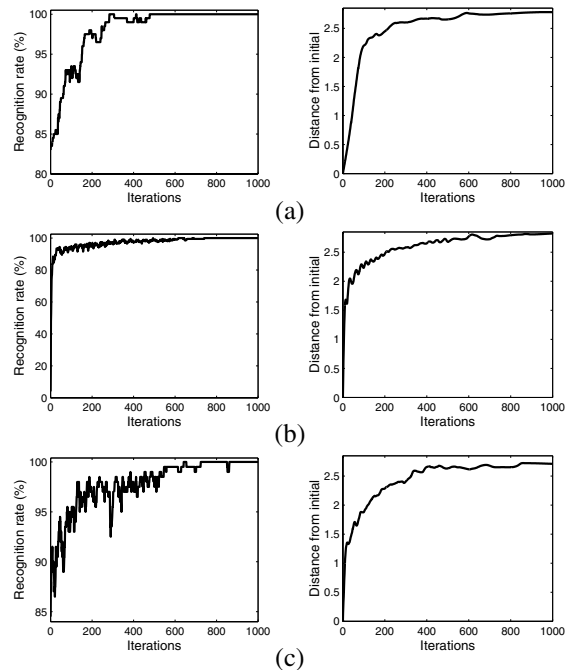


Figure 1: Plots of $F(X_t)$ (left) and distance of X_t from X_0 (right) versus t for different initial conditions. (a) $X_0 = U_{PCA}$, (b) $X_0 = U_{ICA}$, (c) $X_0 = U_{FDA}$. For these curves, $n = 154$, $d = 5$, $k_{train} = 5$, and $k_{test} = 5$.

¹<http://www.uk.research.att.com/facedatabase.html>

We have studied the variation of optimal performance versus the subspace rank denoted by d . We set $k_{train} = 5$, $k_{test} = 5$, and a random value is used for X_0 . It is expected that a larger d leads to a better performance, or makes it easier to achieve a perfect performance. Fig. 2 shows the results for two different values of d . In Fig. 2(a), for $d = 3$, it takes about 2000 iterations for the process to converge to a solution with perfect performance. In Fig. 2(b), for $d = 10$, it takes about 300 iterations.

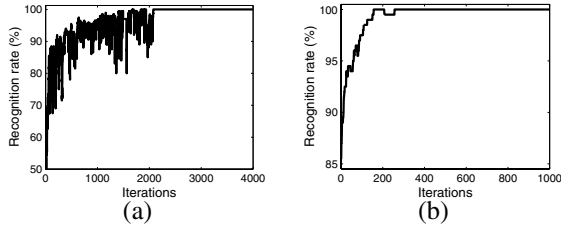


Figure 2: $F(X_t)$ versus t for different values of d . (a) $d = 3$. (b) $d = 10$.

Next, we have studied the optimal performance versus the training size k_{train} . Fig. 3 shows two results with different values of k_{train} . In this experiment, $n = 154$, $d = 5$ and random bases were used as initial conditions. Also, the division of images into training and test sets was random. 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. 3(a) shows the case with $k_{train} = 1$ ($k_{test} = 9$) where it takes about 3000 iterations for the process to converge to a perfect solution. In Fig. 3(b), where $k_{train} = 8$ ($k_{test} = 2$), the process converges to a perfect solution in about 300 iterations.

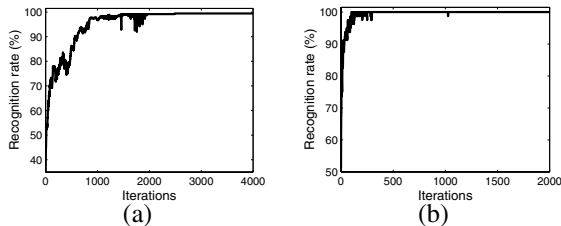


Figure 3: $F(X_t)$ versus t for different divisions of database into training and test sets (keeping $n = 154$, $d = 5$ fixed). (a) $k_{train} = 1$, $k_{test} = 9$. (b) $k_{train} = 8$, $k_{test} = 2$.

We have also studied the algorithmic performance on the COIL dataset, obtaining results similar to ones described earlier for the ORL dataset cases. Fig. 4 shows two representative results. Fig. 4(a) shows a case with random basis as initial condition, $n = 64$, $d = 5$, $k_{train} = 4$, and $k_{test} = 68$. The process converges to an optimal solution with perfect classification. Fig. 4(b) shows a case with $X_0 = U_{ICA}$, $n = 64$, $d = 5$, $k_{train} = 8$, and $k_{test} = 64$.

The process moves very effectively initially as $F(U_{ICA})$ is small. Again it converges to a perfect classification solution.

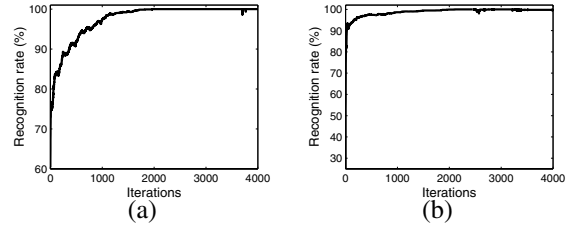


Figure 4: $F(X_t)$ versus t under different settings on the COIL dataset. (a) $k_{train} = 4$, $k_{test} = 68$, $d = 5$, and X_0 is random. (b) $k_{train} = 8$, $k_{test} = 64$, $d = 5$, and $X_0 = U_{ICA}$.

4.2 Performance 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} . This will further emphasize the importance of using optimal representations in algorithms.

Fig. 5(a) shows the recognition performance F (for the ORL database) versus d for four different kinds of subspaces: (i) optimal subspace X^* computed using Algorithm 1, (ii) U_{PCA} , (iii) U_{ICA} , and (iv) U_{FDA} . It shows that the performances from PCA and FDA are similar but the ICA performance is considerably low. We have compared the values of $F(X^*)$ with $F(U_{PCA})$, $F(U_{ICA})$, and $F(U_{FDA})$ for different values of k_{train} and k_{test} , on the ORL database. Figure 5(b) shows the performance of the proposed method as well as standard linear subspace methods with d set at 5. While commonly used subspaces do not perform well especially when k_{train} is small, on the other hand, there exist solutions that give close to perfect recognition performance (99.5%) even when there is only one training image per subject.

The above examples show the possible optimal performance using linear representations when the performance measure can be evaluated. In practice, we often have only a limited number of training images and we are interested in linear subspaces that lead to better performance on the test set which is unknown. To simulate this setting, we have modified Eqn. 9 to be defined using only the available training images, given by,

$$\rho(I_{c,i}, U) = \frac{\min_{c' \neq c, j} d(I_{c,i}, I_{c',j}; U)}{\min_{j \neq i} d(I_{c,i}, I_{c,j}; U) + \epsilon}. \quad (11)$$

Eqn. 11 relates the leave-one-out recognition performance

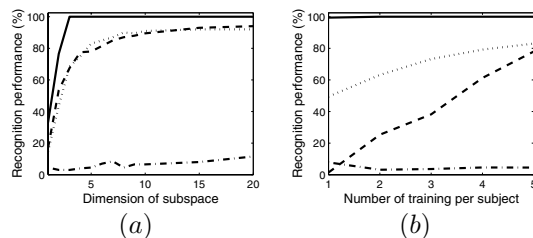


Figure 5: The performance of different linear subspaces with respect to the dimensionality and the number of training images on the ORL dataset. Here solid line is the optimal basis from the gradient search process, dashed line FDA, dotted line PCA, and dash-dotted line ICA. (a) The performance versus d with $k_{train} = 5$. (b) The performance versus k_{train} with $d = 5$.

on the training set. We have applied this modified measure on the ORL dataset by randomly dividing all the images into a non-overlapping training and test set. Fig. 6(a) shows the leave-one-out recognition performance on the training images of the optimal basis along with common linear representations and Fig. 6(b) the corresponding performance on a separate test set. The optimal subspace, found based only on the training set, provides the best performance on the test set under all the cases. This indicates the measure we have used gives better generalization. By imposing additional constraints, we may improve the performance on the test further based on the training set only. This needs further investigation.

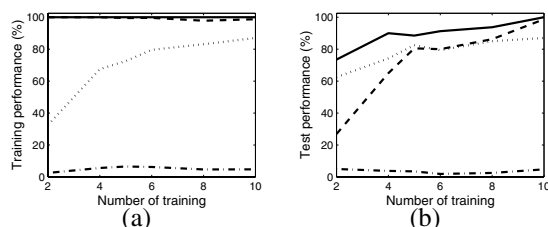


Figure 6: Recognition performance on the training and test sets using the ORL face dataset. See caption of Fig. 5 for legend. (a) The leave-one-out recognition performance on the training set. (b) The recognition performance on a separate test set. In the case of 10 training images per subject, it shows the leave-one-out performance on the training set as the test set is empty.

5 Conclusion

In this paper, we have proposed an MCMC simulated annealing algorithm to find the optimal linear subspaces assuming that the performance function F can be computed. Our extensive experiments demonstrate its effectiveness. This algorithm makes it possible to study and explore the

generalization and other properties of linear representations for recognition systematically, which could lead to significant performance improvement within the linear representation framework.

Acknowledgments: We thank the producers of the ORL and COIL datasets for making them available to the public. This research has been supported in part by the grants NSF DMS-0101429, NMA 201-01-2010, and NSF CCR-9912415.

References

- [1] Y. Amit. A multifold approximation to diffusions. *Stochastic Processes and their Applications*, 37(2):213–238, 1991.
- [2] P. N. Belhumeur, J. P. Hefanpha, 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.
- [3] W. M. Boothby. *An Introduction to Differential Manifolds and Riemannian Geometry*. Academic Press, 1986.
- [4] S. Geman and C.-R. Hwang. Diffusions for global optimization. *SIAM J. Control and Optimization*, 24(24):1031–1043, 1987.
- [5] U. Grenander and M. I. Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society*, 56(3), 1994.
- [6] A. Hyvarinen. Fast and robust fixed-point algorithm for independent component analysis. *IEEE Transactions on Neural Networks*, 10:626–634, 1999.
- [7] A. M. Martinez and A. C. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.
- [8] S. K. Murase and S. K. Nayar. Visual learning and recognition of 3-d objects from appearance. *International Journal of Computer Vision*, vol. 14(1):5–24, 1995.
- [9] C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer Text in Stat., 1999.
- [10] A. Srivastava, U. Grenander, G. R. Jensen, and M. I. Miller. Jump-diffusion markov processes on orthogonal groups for object recognition. *Journal of Statistical Planning and Inference*, 103(1-2):15–37, 2002.
- [11] A. Srivastava, A. B. Lee, E. P. Simoncelli, and S.-C. Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18(1), 2003.