Available online at www.sciencedirect.com

ScienceDirect

Neural Networks 21 (2008) 214–221

Neural Networks

www.elsevier.com/locate/neunet

2008 Special Issue

Learning representations for object classification using multi-stage optimal component analysis[☆]

Yiming Wu[a,*], Xiuwen Liu[a], Washington Mio[b]

[a] Department of Computer Science, Florida State University, Tallahassee, FL 32306, USA
[b] Department of Mathematics, Florida State University, Tallahassee, FL 32306, USA

Received 8 August 2007; received in revised form 2 December 2007; accepted 11 December 2007

Abstract

Learning data representations is a fundamental challenge in modeling neural processes and plays an important role in applications such as object recognition. Optimal component analysis (OCA) formulates the problem in the framework of optimization on a Grassmann manifold and a stochastic gradient method is used to estimate the optimal basis. OCA has been successfully applied to image classification problems arising in a variety of contexts. However, as the search space is typically very high dimensional, OCA optimization often requires expensive computational cost. In multi-stage OCA, we first hierarchically project the data onto several low-dimensional subspaces using standard techniques, then OCA learning is performed hierarchically from the lowest to the highest levels to learn about a subspace that is optimal for data discrimination based on the K-nearest neighbor classifier. One of the main advantages of multi-stage OCA lies in the fact that it greatly improves the computational efficiency of the OCA learning algorithm without sacrificing the recognition performance, thus enhancing its applicability to practical problems. In addition to the nearest neighbor classifier, we illustrate the effectiveness of the learned representations on object classification used in conjunction with classifiers such as neural networks and support vector machines.

Keywords: Dimension reduction; Linear data representations; Feature selection; Object recognition; Object classification; Optimal component analysis

1. Introduction

Learning algorithms for neural network models have been a focal point (Bishop, 1995; Geman & Bienenstock, 1992). Bishop (1995) stated that the choice of pre-processing and feature extraction techniques is "one of the most significant factors in determining the performance of the final system". In the past decades, linear subspace representation methods, such as Principal Component Analysis (PCA) (Jolliffe, 1986; Turk & Pentland, 1991), Independent Component Analysis (ICA) (Comon, 1994; Hyvarinen, Karhunen, & Oja, 2001), Canonical Correlation Analysis (CCA) (Anderson, 2003; Reiter, Donner, Langs, & Bischof, 2006) and Linear Discriminant Analysis (LDA) (Duda, Hart, & Stock, 2000; Zhao, Chellappa, & Phillips, 1994), have been widely used for learning representations suitable for neural networks. For example, Zhu and Yu (1994) implemented a system for face recognition with eigenfaces and a backpropagation neural network. Eleyan and Demirel (2005) proposed a face recognition method in which features are first extracted using PCA and faces are classified using feed-forward neural networks. ICA-based recognition methods, (e.g. Bartlett, Movellen, and Sejnowski (2002) and Kwak and Pedrycz (2007)), tend to give better recognition performance than PCA-based methods as they take high-order statistics of data into account. LDA-based methods, on the other hand, use class information and try to find an optimal basis that maximize the between-class scatter while minimizing the within-class scatter, and are also frequently employed in face and object recognition (Etemad & Chellappa, 1997).

These classical linear representation methods, in general, are not optimal for classification or recognition. For example, PCA and ICA are optimized for data reconstruction and statistical independence, not for the selection of discriminative features. CCA is another multivariate statistical method which extracts

[☆] An abbreviated version of some portions of this article appeared in Wu, Liu, and Mio (2007) as part of the IJCNN 2007 Conference Proceedings, published under IEE copyright.

[*] Corresponding author. Tel.: +1 850 645 2257; fax: +1 850 644 0058.
E-mail addresses: ywu@cs.fsu.edu (Y. Wu), liux@cs.fsu.edu (X. Liu), mio@math.fsu.edu (W. Mio).

0893-6080/$ - see front matter © 2007 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2007.12.023

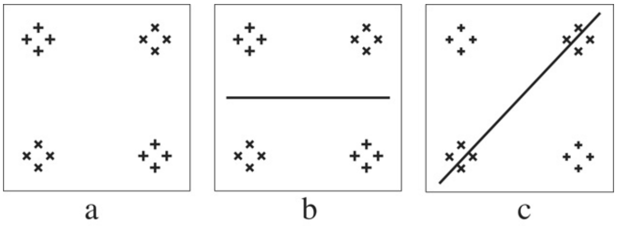

Fig. 1. A synthetic dataset consisting of two classes, each with two clusters of four points: (a) the data set of two classes ('+' and '×') with eight points each in $\Re^2$; (b) the one-dimensional subspace obtained from PCA, ICA, and LDA; (c) a one-dimensional optimal subspaces representation obtained using OCA.

the most coherent features among two data channels. LDA assumes that the conditional probability distribution of each class is Gaussian with the same variance. As the distributions of real images are typically non-Gaussian (e.g. Srivastava, Liu, and Grenander (2002)), in recognition tasks, there is no theoretic guarantee of optimality of LDA basis. This is also evident in comparative studies reported in the literature (e.g. Belhumeour, Hespanha, and Kriegman (1997) and Martinez and Kak (2001)). In fact, one can construct examples in which all the common choices of learning algorithms give the worst possible performance. Such an example is shown in Fig. 1, which consists of two classes ('+' and '×') with eight points each, and the points are presented in clusters of four. It can be shown that the one-dimensional subspace resulting from PCA, ICA, and LDA coincides with either the horizontal or the vertical axis. If we use the nearest neighbor classifier and let a point from each cluster be used for training, the one-dimensional basis obtained from PCA, ICA, and LDA gives the worst performance.

It is thus apparent that, in the context of object recognition, a more relevant question is that of finding a linear representation that optimally selects discriminating features. Unlike the classical methods, the recently proposed Optimal Component Analysis (OCA) (Liu, Srivastava, & Gallivan, 2004; Srivastava & Liu, 2005) provides a general optimality criterion. The search for optimal linear representations, or an optimal subspace, is based on a stochastic optimization process which maximizes a pre-specified performance function over all subspaces of a particular dimension and is estimated using a Markov Chain Monte Carlo (MCMC) type algorithm. OCA exhibits good performance on face and object recognition. Fig. 1(c) shows an optimal subspace representation obtained by the OCA method.

The stochastic search techniques employed in OCA typically result in heavy computational costs, which limits the applicability of OCA to practical problems that involve feature extraction and object recognition. As an example, consider a facial recognition experiment based on the ORL data set (Samaria & Harter, 1994). OCA learning takes approximately one day to run 1000 iterations to estimate an optimal subspace. Obviously, this is not practical for most object recognition applications. In our previous work, a two-stage strategy was proposed to address this problem (Wu, Liu, Mio, & Gallivan, in press). In this approach, the input data is first reduced to a lower dimension using methods such as PCA or LDA; then, the OCA search is performed in the reduced space. This strategy leads to significant computational gains. However, it is generally difficult to determine a good choice for the reduced subspace. In this paper, a multi-stage strategy is proposed to address this problem. The idea of multi-stage OCA (M-OCA) was presented in a previous short paper (Wu et al., 2007): the data is first hierarchically reduced into several levels using shrinkage matrices; then, the OCA search is performed hierarchically from the lowest to the highest levels. The basis is expanded progressively from the optimal basis obtained in the previous level. As the learning process of each level starts with a good initial selection from the previous level, M-OCA achieves good recognition performance. Also, since the dimensions of the Grassmann manifolds at the lower levels are much smaller than that of the Grassmannian in the original space, M-OCA reduces the computational costs associated with the original algorithm significantly, thus making OCA learning feasible in applications.

The rest of the paper is organized as follows: Section 2 gives a brief review of OCA and the proposed M-OCA method is presented in Section 3; A comprehensive study of the performance of the M-OCA algorithm is presented in Section 4; Section 5 concludes the paper with a summary and a discussion of future work.

2. Optimal component analysis

Optimal Component Analysis is a dimension reduction technique designed to find an optimal subspace (of a prescribed dimension) of feature space that optimizes the ability of the nearest neighbor classifier to index and classify images or other data. The measurement of optimality is based on training data and the algorithm yields an orthonormal basis of the estimated optimal subspace. More specifically, let $U \in \Re^{n \times d}$ be a matrix whose columns form an orthonormal basis of a d-dimensional subspace of $\Re^n$, where n is the size of the input image and d is the dimension of the desired subspace (generally $n \gg d$). For an image I, viewed as an n-vector, the vector of coefficients is given by $\alpha(I, U) = U^T I \in \Re^d$ and represents the orthogonal projection of I onto the subspace S_U spanned by the columns of U. Suppose the training data consists of representatives of C classes of images, with each class represented by k_{train} images denoted by $I_{c,1}, \ldots, I_{c,k_{\text{train}}}$, where $c = 1, \ldots, C$. Let

$$\rho(I_{c,i}, U) = \frac{\min\limits_{c' \neq c, j} D(I_{c,i}, I_{c',j}; U)}{\min\limits_{j \neq i} D(I_{c,i}, I_{c,j}; U) + \epsilon}. \tag{1}$$

The numerator is the distance from $I_{c,i}$ to the closest training image not in its class and the denominator is the distance from $I_{c,i}$ to the closest training image in the same class. Here, D denotes Euclidean distance; that is,

$$D(I_1, I_2; U) = \|\alpha(I_1, U) - \alpha(I_2, U)\|, \tag{2}$$

where $\|\cdot\|$ is the usual 2-norm. In Eq. (1), $\epsilon > 0$ is a small number introduced to avoid division by zero. Note that large values of ρ are desirable, since this means that $I_{c,i}$ will be closer to its class than to other classes after projection onto the subspace S_U. A performance function F is defined to essentially measure the average value of ρ over all training images, as follows:

$$F(U) = \frac{1}{Ck_{\text{train}}} \sum_{c=1}^{C} \sum_{i=1}^{k_{\text{train}}} h(\rho(I_{c,i}, U) - 1), \tag{3}$$

where $h(\cdot)$ is a monotonically increasing bounded function used to control bias with respect to particular classes in measurements of performance. In our implementation, we use $h(x) = 1/(1 + \exp(-2\beta x))$, where β is a parameter that controls the degree of smoothness of $F(U)$. Thus, F is a quantifier of the ability of the nearest neighbor classifier to discern the C classes after projection onto S_U. Moreover, as $\beta \to \infty$ and $\epsilon \to 0$, F gives precisely the recognition performance of the nearest neighbor classifier after projection to the subspace given by U (see Liu et al. (2004)).

Under this formulation, $F(U) = F(UH)$ for any $d \times d$ orthogonal matrix H since $D(I_1, I_2; U) = D(I_1, I_2; UH)$. The Grassmann manifold, $\mathcal{G}(n, d)$, is the set of all d-dimensional subspaces of $\Re^n$. It is a compact, connected manifold of dimension $d(n-d)$, which can be represented either by a basis (non-uniquely) or by a projection matrix (uniquely). Choosing the former, let U be an $n \times d$ matrix whose columns are an orthonormal basis for the given subspace of $\Re^n$ and let $[U]$ denote the set of all the orthonormal bases of span(U), i.e., $[U] = \{UH | H \in \Re^{d\times d}, H^T H = I_d\} \in \mathcal{G}(n, d)$. The value of $F(U)$ depends only on $[U]$; unlike the actual recognition performance, $F([U])$ is smooth and thus allows us to use gradient type algorithm to solve the optimization problem. An optimal d-dimensional subspace for the given classification problem from the viewpoint of the training data is given by

$$[\hat{U}] = \arg \max_{[U]\in\mathcal{G}_{n,d}} F([U]). \tag{4}$$

To solve this optimization problem, Liu et al. (2004) present an algorithm utilizing the geometric properties of the Grassmann manifold. A Monte Carlo version of a stochastic gradient algorithm that uses simulated annealing is used to find an optimal subspace.

A brute force implementation of OCA is typically computationally expensive and may limit its applicability. The computational complexity C_n of each iteration of this algorithm is

$$C_n = O(d \times (n - d) \times k_{\text{train}}^2 \times n \times d). \tag{5}$$

C_n is obtained from the following analysis. The dimension of the gradient vector is $d \times (n - d)$. For each dimension and for each training image, the closest images in all the classes need to be found to compute the ratio in Eq. (1) and to the performance F in Eq. (3), which gives the factor k_{train}^2. The term $n \times d$ comes from Eq. (2). Therefore, we obtain the above estimate for each iteration. The overall computational complexity is $C_n \times T$ where T is the number of iterations. Thus, we see that the computation at each iteration depends on several factors and the complexity is $O(n^2)$. In typical applications, n is relatively large since it represents the number of pixels in an image. This leads to an algorithm that can be very time consuming.

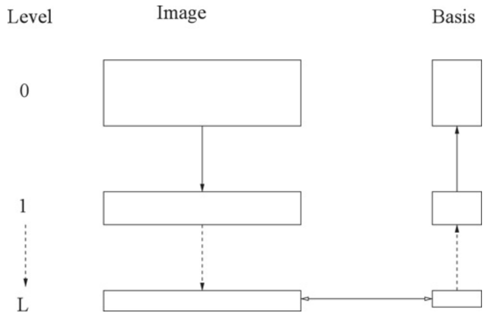

Fig. 2. Multi-stage search process. Firstly, an optimal basis U_L is obtained at level L through an OCA search on $\mathcal{G}_{n_L,d}$. Then, a basis $\bar{U}_{L-1}$ at level $L-1$ is obtained by expanding U_L, where $\bar{U}_{L-1}$ is used to initialize the learning at level $L-1$ on $\mathcal{G}_{n_{L-1},d}$. This search – expand-basis – process is iterated until we get an optimal basis at level 0.

The high computational cost of OCA motivated the development of two-stage OCA (Wu et al., in press). Instead of solving the optimization in the original space, by limiting the search to (subspaces) of the span of the training images, we showed that we can achieve better efficiency while preserving effectiveness. In this paper, we further enhance the strategy by developing a technique that we refer to as M-OCA.

3. Multi-stage OCA

In the first stage of two-stage OCA (Wu et al., in press), the dimension of the input data is reduced using well-known dimension reduction methods such as PCA, ICA, LDA and QR factorization. Subsequently, an OCA search is performed on the lower-dimensional space. Obviously, as the OCA search space is smaller, the search time can be greatly reduced. However, two-stage OCA requires the selection of a dimension in the first stage. Experiments indicate that there is an optimal range of values: if the dimension is too large (in the extreme case, it becomes the original OCA), the complexity increases and the search becomes ineffective; on the other hand, if it is too small (in the extreme case, the second stage is not required and it degenerates to the first stage), effective linear representations for discrimination may not be achievable. M-OCA, on the other hand, uses a multi-level learning strategy and effectively solves the problem.

3.1. Multi-stage learning

The learning process of M-OCA is illustrated in Fig. 2. First, we chose the number $L + 1$ of levels and shrinkage matrices $A_l \in \Re^{n_l \times n_{l+1}}$, $0 \le l < L$, usually constructed with classical techniques such as PCA or the K-Means algorithm (see e.g. Zhang and Liu (2003), Forsyth and Ponce (2003) and MacQueen (1967)). Level 0 is viewed as the highest level and level L as the lowest. Then, we recursively shrink training image data set $I_l \in \Re^{K_{\text{train}} \times n_l}$, $0 \le l < L$, to get n_{l+1}-dimensional image data I_{l+1} via right multiplication by the shrinkage matrix A_l at each level. If we denote the shrinkage factor as m_l and the dimension of the image data at level l as n_l, then $n_l = \frac{n_{l-1}}{m_l} = \frac{n_0}{\prod_{i=1}^{l} m_i}$.

Our goal is to find an optimal basis $\hat{U}_0 \in \Re^{n_0 \times d}$ at level 0. To accomplish this, we hierarchically search an optimal basis at each level. The search begins from level L on the Grassmann manifold $\mathcal{G}_{n_L,d}$ with image data I_L of dimension n_L. The search can be performed efficiently since the learning space $\mathcal{G}_{n_L,d}$ is relatively low dimensional. After getting a basis $\hat{U}_L$ of an optimal d-dimensional subspace at level L, we obtain a preliminary basis $\bar{U}_{L-1}$ at level $L - 1$ by expanding U_L through left multiplication by the shrinkage matrix A_{L-1}. Then, we use the basis $\bar{U}_{L-1}$ to initialize a new OCA search at level $L - 1$ on $\mathcal{G}_{n_{L-1},d}$, with image data $I_{L-1} \in \Re^{n_{L-1}}$. As the recognition performance based on $\bar{U}_{L-1}$ is consistent with that of $\hat{U}_L$, the search at this level will be significantly faster than it would have been just using standard OCA since it is initialized with a high-performance basis. This process is repeated until we reach level 0. At this point, we will have a basis that is "optimal" enough for discrimination, as shown in various experiments. In summary, the search is performed from the lowest level L to the highest level 0; the lower the level, the more efficient the search. The search result at a lower level provides a good initialization for the next level improving the efficiency of OCA without compromising its discriminative power. The recognition performance keeps on increasing at each level. The pseudo-code for this procedure is given in Algorithm 1.

Algorithm 1. M-OCA Algorithm

Input: Training image data set matrix $I_{\text{train}} \in \Re^{K_{\text{train}} \times n_0}$, shrinkage factors $m_1, \ldots, m_L$.
Output: Optimal basis $\hat{U}_0$ of level 0

```
1. For $l = 0, \ldots, L - 1$
   BEGIN
     (a) Using dimension reduction methods, such as PCA or
   the K-Means algorithm, construct shrinkage matrices $A_l$,
   $0 \le l < L$, where $A_l \in \Re^{n_l \times n_{l+1}}$ and $n_{l+1} = \frac{n_l}{m_l}$.
     (b) Set $I_{l+1} = I_l A_l$.
   END
2. Learn an optimal basis $U_L$, at level $L$, by doing an OCA
   search on $\mathcal{G}_{n_L,d}$, where $n_L$ is the dimension of the image data
   at level $L$.
3. For $l = L - 1, \ldots, 1$,
   BEGIN
     (a) Set $\bar{U}_l = A_l \hat{U}_{l+1}$,
     (b) Using $\bar{U}_l$ as the initial basis, search for an optimal
   basis $\hat{U}_l$ at level $l$ employing an OCA search on $\mathcal{G}_{n_l,d}$ with data
   size $n_l$.
   END
4. At level 0, set $\hat{U}_0 = A_0 \hat{U}_1$.
```

From the discussion in Wu et al. (2007), PCA and K-means are suitable choices for obtaining the data shrinkage matrices for M-OCA. Also, since the performance in the initial iteration of a higher level is consistent with the performance in the final iteration of the previous lower level, the recognition performance is improved level-by-level and a good recognition performance can be achieved at level 0.

3.2. Computational analysis

Here, we estimate some of the computational gains realized by using the M-OCA algorithm. For each iteration, the computational complexity with images of size n_0 is $C_{n_0} = O(d \times (n_0 - d) \times k_{\text{train}}^2 \times n_0 \times d)$ and the complexity with images of size n_l is

$$\begin{aligned} C_{n_l} &= O(d \times (n_l - d) \times k_{\text{train}}^2 \times n_l \times d) \\ &= O\left(\frac{(n_l - d) \times n_l}{(n_0 - d) \times n_0} C_{n_0}\right) \end{aligned}$$

and the total computational complexity will be of the order of

$$\begin{aligned} C_{\text{total}} &= \sum_{l=1}^{L} C_{n_l} \\ &= \sum_{l=1}^{L} \frac{(n_l - d) \times n_l}{(n_0 - d) \times n_0} C_{n_0} \\ &= \sum_{l=1}^{L} \frac{(n_0 - \prod_{i=1}^{l} m_i d)}{(n_0 - d)(\prod_{i=1}^{l} m_i)^2} C_{n_0}. \end{aligned} \tag{6}$$

In applications, we usually select $2 \le L \le 5$, $10 < d < 100$ and $m_i > 1$, so that $C_{\text{total}} \ll C_{n_0}$. Our experiments demonstrate that, in practice, an improvement of this magnitude is actually realized in this range of dimensions, so that the multi-stage version is much more efficient than the original OCA.

4. Experimental results

We evaluate the effectiveness of the M-OCA in this section. The data sets used in our experiments are described in Section 4.1. In 4.2, we present a set of experiments to test the recognition accuracy and efficiency of M-OCA algorithm using the K-nearest neighborhood classifier. In Section 4.3, we compare the recognition performance of M-OCA with PCA using neural network and SVM classifiers. Speed is measured on a workstation with an Intel Xeon 3.00 GHz CPU and 8.0G RAM.

4.1. Data sets

The ORL, PIE, AR face, and COIL object data sets are used in our experiments. Fig. 3 shows some sample images from these data sets.

- ORL face data set (Samaria & Harter, 1994). It contains 400 face images of 40 individuals. The image size is 92 × 112. The face images are perfectly centralized. The major challenge on this data set is the variation of the face pose. There is no lighting variation with minimal facial expression variation and no occlusion. We use the whole image as an instance, i.e., the dimension of an instance is 92 × 112 = 10,304.

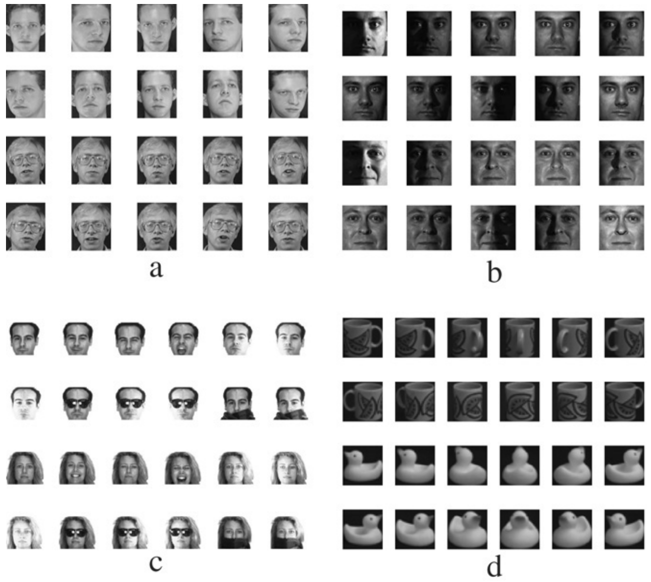

Fig. 3. Some example images of data sets used in experiments. (a): ORL face data set, which contains 40 classes, each has 10 images; (b): PIE face data set, which contains 66 classes, each class has 21 images; (c): AR face data set, which contains 156 classes, each class has 13 images; (d): COIL data set, which contains 100 objects, each has 72 images.

- PIE face data set (Sim, Simon, & Bsat, 2001). It contains 66 persons with 21 images each. Images of each person was taken under 13 different poses, 43 different illumination conditions and 4 different expressions. We use those 676 images that are cropped manually. The image size is 100 × 100 = 10,000.
- AR face data set (Martinez & Benavente, 1998). It is a large face image data set. An instance of a face may contain significantly large areas of occlusion, due to the presence of sun glasses and scarves. The existence of occlusion dramatically increases the within-class variations of AR face image data. In this study, we use a subset of AR containing 1638 face images of 126 individuals. Its image size is 768 × 576. We first crop the image from the row 100–500 and the column 200–550, and then sub-sample the cropped images with sample step 4 × 4. The dimension of each instance is reduced to 101 × 88 = 8888.
- COIL object data set (Nayar, Nene, & Murase, 1996). The COIL-100 data set consists of color images of 100 objects where the images of the objects that were taken at pose intervals of 5 degrees, i.e., 72 poses per object. The images were also normalized such that the larger of the two object dimensions (height and width) fits the image size of pixels. The image size is 32 × 32 = 1024.

4.2. Recognition performance of multi-stage OCA

The first set of experiments evaluates the performance of M-OCA algorithm in terms of recognition accuracy and efficiency. Here, we use half of the data in each class for training and another half for testing. In all the following experiments, PCA is used to compute the shrinkage matrix. The shrinkage level is 3 and the subspace dimension d in each level is set to 10.

Table 1
Data dimension at each level

Level	0	1	2	3
ORL	10,304	199	100	50
PIE	10,000	600	100	20
AR	8,888	2000	1000	200
COIL	10,000	500	100	20

Table 2
Recognition accuracy (%) on test images of M-OCA for K-NN classifier

Dataset	1-NN	3-NN	4-NN	5-NN
ORL	100	98.50	98.00	95.50
PIE	100	99.17	98.35	94.36
AR	97.95	95.24	94.90	92.97
COIL	99.41	98.36	95.86	94.75

Table 1 shows the image size of each level, where level 0 is the original image space. OCA search is performed in each level except level 0 for 500 iterations. We also set a stop criterion for the M-OCA search, that is, we claim that the optimal basis is obtained when the recognition accuracy reaches 100%, or improvement of recognition accuracies in two adjacent levels is smaller than 1%. Here the K-Nearest Neighborhood (KNN) is used as classifier. Table 2 gives the classification accuracy of the M-OCA algorithm with different KNN on ORL, PIE, AR and COIL data sets. From the table, we have the following important observations:

- KNN with $K = 1$ usually performs the best by M-OCA on these four data sets. There is a clear trend of decrease in accuracy for each data set as K increases.
- M-OCA gives good classification accuracy on ORL and PIE data sets. For example, the M-OCA algorithm can achieve 100% on ORL and PIE data sets. This is mainly due to the relatively small within-class variations in these data.
- For the COIL data sets, although orientation difference exists between each image in the same object. However, M-OCA can still achieve better classification results than other methods (e.g. Roth, Yang, and Ahuja (2002)).
- The classification accuracy on AR data set is slightly worse than that on other data sets. This is mainly because the images contain large areas of occlusion whose direct consequence is large within-class variation of each individual. However, the classification results give marginal improvement when compared with other classification methods (e.g. Lu and Jain (2003)).

We also compare the result of M-OCA with another non-linear representation method—Kernel Discriminant Analysis (KDA). KDA can find an optimal nonlinear representation in which discrimination of the training samples is optimized. Table 3 gives the comparative classification accuracies of M-OCA and KDA on these four data sets, using 1-NN classifier. It shows that M-OCA have better classification accuracy in all these data sets.

Fig. 4 shows the evolution of recognition accuracy and performance function of M-OCA algorithm on the data sets.

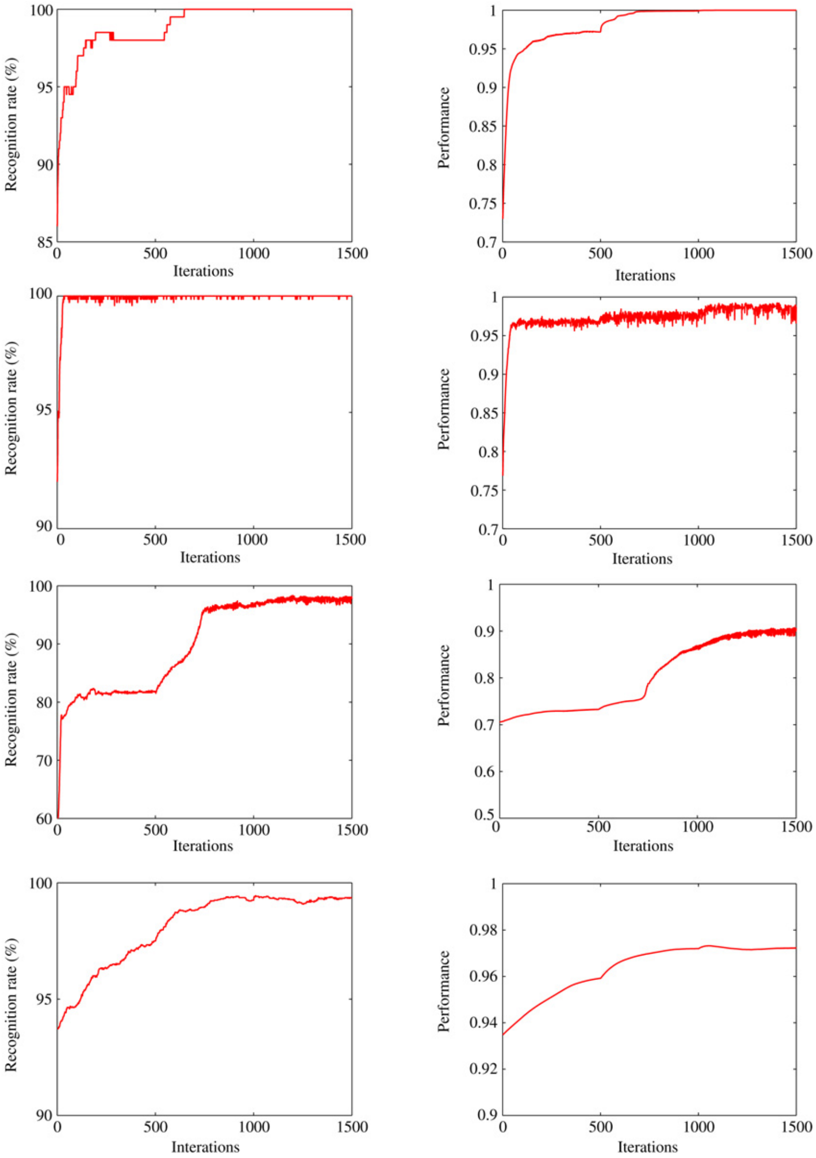

Fig. 4. Evolution of recognition accuracy and performance function F on test data sets. Left column: Y-axis: recognition accuracy, X-axis: iteration numbers; Right column: Y-axis: performance function F, X-axis: iteration numbers. From top to bottom: ORL, PIE, AR, COIL.

Table 3
Recognition accuracy (%) on test images of M-OCA and KDA for 1-NN classifier

Dataset	M-OCA	KDA
ORL	100	97.50
PIE	100	98.48
AR	97.95	91.24
COIL	99.41	98.36

Table 4
Time (in seconds per iteration) and classification accuracy (%) comparison of M-OCA and original OCA

Data set	Measured items	M-OCA	Original OCA
ORL	Time	2.14	173
	Accuracy	100	100
PIE	Time	10.44	224
	Accuracy	100	100
AR	Time	46.40	421
	Accuracy	97.34	98.03
COIL	Time	46.12	531
	Accuracy	99.11	99.41

Although we can set a stop criterion for OCA searching in each level, in order to better illustrate the evolution of performance of the M-OCA, we set the level to 3 and run 500 iterations in each level. The left figures in each row show the evolution of recognition accuracy of test data. We can see that in each level the recognition accuracy is increased and the recognition accuracy of final iteration in a higher level is consistent with the initial point of next lower level. The right figure in each row shows the evolution of performance function F, which is also convincing.

Table 4 shows the running time and classification accuracy of M-OCA obtained in this experiment. We run OCA search 500 iterations in each level. From this table, we can see that

Table 5
Class number and subspace selected in the second set of experiments

Dataset	# of class	subspace
ORL	40	10
PIE	66	10
AR	31	20
COIL	25	20

the running time of M-OCA is greatly reduced compared to the original OCA algorithm; however, the classification accuracy is comparable with the original OCA algorithm. Reader can refer to Wu et al. (2007) for further details.

4.3. Performance evaluation using BP Neural Networks and SVM classifiers

Neural networks (Rumelhart, Hinton, & Williams, 1986) and Support Vector Machines (SVM) (Vapnik, 1995) can be trained to approximate complex functions in various fields of applications including pattern recognition and object classification. Instead of using K-Nearest Neighborhood as classifier, backpropagation neural networks (BPNN) and SVM classifiers are used in this experiment. We use a 3 layer backpropagation neural network for all cases. The number of nodes in the input layer is determined by the length of the feature of data, which is determined by the subspace dimension d; the number of nodes in the output layer is determined by the number of class in each data set; and the number of nodes in the hidden layer is set to 20. The number of class and subspace selected for the set of experiments are listed in Table 5. We use the whole data sets for ORL and PIE data set, for AR and COIL data sets, we select part of the data set to make neural networks converge quickly during training.

Support vector machines map input vectors to a higher-dimensional space where a maximal separating hyperplane is constructed. Two parallel hyperplanes are constructed on each side of the hyperplane that separates the data. In our experiment, one-vs-one strategy (Schölkopf, Burges, & Smola, 1998) is used to solve the multi-class classification problem. It forms a binary classifier for each class-pair and thus $C(C-1)/2$ classifiers are required. For the test input, the decision is made by combining these $C(C-1)/2$ classifier outputs using majority voting. A Gaussian kernel is used for all SVM classifiers.

Table 6 shows the classifier result of M-OCA and PCA using BPNN and SVM classifiers. We tested the classification performance using PCA and M-OCA for representation learning. We can see that for the ORL, PIE and COIL data sets, M-OCA has significant improvement for both neural networks and SVM classifiers. For AR data set, it also gives improvement, although the improvement is smaller.

5. Discussion and future work

In this paper, we have proposed a M-OCA algorithm which extends the two-stage OCA algorithm by first projecting the data to several different lower-dimensional levels using shrinkage matrices. Then, OCA is performed hierarchically from the lowest to the highest levels. After constructing an optimal basis at a level, we expand the basis to a higher level and use this expanded basis to initialize the OCA search at the next higher level. This strategy greatly reduces the OCA search time, while essentially preserving the recognition performance. The nature of OCA learning, which involves a stochastic optimization, allows us to utilize a multi-stage strategy to improve the computational efficiency. Several experimental results using K-NN, Neural networks and SVM classifiers

Table 6
Recognition accuracy(%) on test images of multi-stage OCA and PCA using BPNN and SVM classifiers

Dataset	# of trainings per class	# of tests per class	PCA/BPNN	M-OCA/BPNN	PCA/SVM	M-OCA/SVM
ORL	2	8	64.06	69.38	69.69	78.75
	3	7	76.07	82.50	89.29	91.07
	5	5	83.00	88.00	93.00	96.50
	7	3	88.33	96.67	94.17	96.67
	8	2	92.50	97.50	95.00	100
PIE	4	17	90.02	92.87	90.26	93.76
	7	14	95.45	96.53	96.10	97.94
	10	11	95.59	98.48	96.56	98.07
	14	7	96.10	98.70	96.67	98.70
	17	4	96.96	99.62	98.86	100
AR	2	11	53.37	54.55	66.57	68.62
	4	9	78.13	79.93	85.30	88.17
	6	7	70.51	70.97	88.94	91.24
	9	4	93.54	94.35	96.67	98.38
	11	2	95.16	96.77	96.77	98.38
COIL	12	60	90.57	94.92	96.43	96.43
	24	48	91.33	97.33	97.50	98.50
	36	36	92.11	98.28	97.72	99.33
	48	24	87.83	97.75	97.41	99.75
	60	12	94.50	99.50	97.83	99.83

demonstrate the improvement achieved using the proposed learning method.

While this paper focuses on linear representations, the proposed M-OCA algorithms can be generalized to model nonlinearity using kernel methods. As shown in Liu and Mio (2005), nonlinear representations induced by a kernel function can be written as linear representations with respect to a basis that depends only on the kernel function and the training set. As multi-stage OCA algorithms significantly enhances the applicability of OCA, kernel analogues of M-OCA should offer similar benefits; this is being investigated.

Acknowledgments

This research was supported in part by NSF grants CCF-0514743 and IIS-0307998. The authors would like to thank the producers of the ORL, PIE, AR and COIL data sets for making them public.

References

Anderson, T. W. (2003). *An introduction to multivariate statistical analysis* (3rd ed.). John Wiley & Sons.

Bartlett, M. S., Movellen, J. R., & Sejnowski, T. J. (2002). Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, *13*, 1450–1464.

Belhumeour, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*, 711–720.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press.

Comon, P. (1994). Independent component analysis, a new concept? *Signal Processing*, *36*, 287–314.

Duda, R. O., Hart, P. E., & Stock, D. (2000). *Pattern classification*. John Wiley & Sons.

Geman, S., & Bienenstock, E. (1992). Neural network and the bias/variance dilemma. *Neural Computations*, *4*, 1–58.

Eleyan, A., & Demirel, H. (2005). Face recognition system based on PCA and feedforward neural networks. *Computational Intelligence and Bioinspired Systems*, *3512*, 935–942.

Etemad, K., & Chellappa, R. (1997). Discriminant analysis for recognition of human face images. *Journal of the Optical Society of America A: Optics, Image Science, and Vision*, *14*, 1724–1733.

Forsyth, D. A., & Ponce, J. (2003). *Computer vision, a modern approach*. Prentice Hall, pp. 315–317.

Hyvarinen, A., Karhunen, J., & Oja, E. (2001). *Independent component analysis*. John Wiley & Sons.

Jolliffe, I. T. (1986). *Principal component analysis*. Springer-Verlag.

Kwak, K. C., & Pedrycz, W. (2007). Face recognition using an enhanced independent component analysis approach. *IEEE Transactions on Neural Networks*, *18*, 530–541.

Liu, X., & Mio, W. (2005). Kernel methods for nonlinear discriminative analysis. In *Proceedings of the international workshop on energy minimization methods in computer vision and pattern recognition* (pp. 584–599).

Liu, X., Srivastava, A., & Gallivan, K. A. (2004). Optimal linear representation of images for object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 662–666.

Lu, X., & Jain, A.K. (2003). Resampling for face recognition. In *AVBPA03* (pp. 869–877).

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th berkeley symposium on mathematical statistics and probability* (pp. 281–297). Berkeley: University of California Press.

Martinez, A.M., & Benavente, R. (1998). The AR face database. CRC Technical Report #24. 1998.

Martinez, A. M., & Kak, A. C. (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*, 228–233.

Nayar, S.K., Nene, S.A., & Murase, H. (1996). Columbia object image library (COIL-100). Tech. Rep. CUCS-006-96.

Reiter, H., Donner, R., Langs, G., & Bischof, H. (2006). Estimation of face depth maps from color textures using canonical correlation analysis. In Computer Vision Winter Workshop (pp. 6–8).

Roth, D., Yang, M., & Ahuja, N. (2002). Learning to recognition objects. *Neural Computation*, *14*, 1071–1104.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by backpropagation errors. *Nature*, *323*, 533–536.

Samaria, F., & Harter, A. (1994). Parameterization of a stochastic model for human face identification. In *Proceedings of 2nd IEEE workshop on applications of computer vision*. FL: Sarasota.

Schölkopf, B., Burges, C. J. C., & Smola, A. J. (1998). *Advances in kernel methods: Support vector machine*. The MIT Press.

Sim, T., Simon, B., & Bsat, M. (2001). The CMU pose, illumination, and expression (PIE) database of human faces. Tech. Report, CMU-RI-TR-01-02. Pittsburgh, PA. Robotics Institute, Carnegie Mellon University.

Srivastava, A., & Liu, X. (2005). Tools for application-driven dimensional reduction. *Neuro Computation*, *67*, 136–160.

Srivastava, A., Liu, X., & Grenander, U. (2002). Universal analytical forms for modeling image probabilities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 1200–1214.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuro-science*, *3*, 71–86.

Vapnik, V. (1995). *The nature of statistical learning theory*. Springer-Verlag.

Wu, Y., Liu, X., Mio, W., & Gallivan, K.A. (2007). Two-stage optimal component analysis via dimensional reduction. Computer Vision and Image Understanding (in press).

Wu, Y., Liu, X., & Mio, W. (2007). Multi-stage optimal component analysis. In *Proceedings of the international joint conference on neural network*. Orlando, FL.

Zhang, Q., & Liu, X. (2003). Hierarchical learning of optimal linear representations. In *Proceedings of the international joint conference on neural networks* (pp. 2247–2252).

Zhao, W., Chellappa, R., & Phillips, P. (1994). Subspace linear discriminant analysis for face recognition. Center for Automation Research. University of Maryland. College Park. Technical Report CAR-TR-914.

Zhu, J., & Yu, Y.L. (1994). Face recognition with eigenfaces. In *Proceeding of the IEEE international conference on industrial technology* (pp. 434–438).