Linear Representation Learning Using Sphere Factor Analysis

Yiming Wu, Xiuwen Liu
Department of Computer Science
Florida State University
Tallahassee, FL, 32306, USA
{ywu, liux}@cs.fsu.edu

Washington Mio
Department of Mathematics
Florida State University
Tallahassee, FL, 32306, USA
mio@math.fsu.edu

Abstract

Representation learning is a fundamental challenge for feature selection and plays an important role in applications such as dimension reduction, data mining and object recognition. Traditional linear representation methods, such as principal component analysis (PCA), independent component analysis (ICA) and linear discriminate analysis (LDA), have good performance on certain applications based on corresponding criteria. However, these linear representation methods are not optimal in general. Sphere factor analysis (SFA) is a recently proposed method which provides a general framework for optimization problems. In term of object recognition, SFA seeks to optimize the discriminant ability of the nearest neighbor classifier for data classification and labeling. Based on the geometry structure of the search space, a gradient search algorithms have been applied to obtain an optimal basis. A detail presentation of these algorithm is given in this paper. Furthermore, to speed up the search procedure of SFA, a two-stage strategy is proposed, which we called two-stage SFA. We illustrate the effectiveness of the original SFA and two-stage SFA methods on UCI data sets and two face data sets.

1. Introduction

Linear representation methods, such as Principle Component Analysis (PCA)[5], Independent Component Analysis (ICA) [1], and Linear Discriminant Analysis (LDA) [3], are attractive due to their relative simplicity. However, drawbacks exist when these methods are applied to the classification problem. For example, PCA is not optimal as it does not utilize the class information. ICA suffers from computational expensiveness, which limits its application to high-dimension data classification. The LDA algorithm is optimal if all class distribution are Gaussian with a single shared covariance which is rarely held in real data.

In the recent years, several novel linear representation methods based on K-nearest-neighbor (KNN) classifier have been proposed, which include Neighborhood Component Analysis(NCA) [4], Optimal component analysis(OCA) [7], etc. NCA considers the probability of each point to select another point as its neighborhood as inherits its labels. The optimal basis is solved using a gradient search method. OCA finds the optimal basis by separating each point far away from points in different class but close to points in the same class so that KNN can achieve good performance. The optimal basis is solved using a gradient search method on a Grassmann manifold. Sphere Factor Analysis (SFA) is another recently proposed KNN based method for simultaneous dimension reduction and optimal feature selection [6]. Given a input data $I \in \Re^n$ and projected subspace $d \ll n$, the goal of SFA is to find a linear transformation $A : \Re^n \rightarrow \Re^d$ that optimizes the discriminative ability of the KNN classifier on the data transformed by A. The idea is that the optimal mapping A will reduce dimension and reconstruct the data so that it becomes more amenable to classification.

The performance function F used to measure "optimality" in sphere factor analysis is similar to that adopted in OCA, but the optimization is carried out over all linear mapping $A : \Re^n \rightarrow \Re^d$, not just orthogonal projections onto subspaces. $F(A)$ quantifies how well suited a linear mapping A is for the classification task at hand. F has the property that it is (nearly) scale invariant reflecting the fact that scaling a data set does not change decisions based on KNN classifier. Thus it suffices to consider linear mapping of unit norm; that is, to optimize F over the unit sphere S in $\Re^{n\times d}$. Moreover, the search space of SFA is on a sphere, a much more simple geometry than the Grassmann manifold of OCA. Thus, significant computational gains in the learning process can be achieved.

To further reduce the computational cost of SFA, a two-stage SFA algorithm is proposed in this paper. In the first stage, we project the data dimension into a lower subspace using some traditional dimension reduction methods, such as PCA, LDA or ICA. In the second stage, the SFA is per-

978-0-7695-3926-3/09 $26.00 © 2009 IEEE
DOI 10.1109/ICMLA.2009.127

formed in the second stage. As the search process is on a much smaller space, the computational cost will be greatly reduced.

The rest of the paper is organized as follows: Section 2 gives a detail discussion of SFA method and the gradient search algorithms are presented in Section 3. In Section 4, a two-stage SFA method is proposed to reduce the computational cost of SFA. A comprehensive study of the performance of SFA and two-stage SFA algorithm is presented in Section 5. The paper is summarized in Section 6.

2 Sphere Factor Analysis (SFA)

Sphere factor analysis (SFA) is a linear feature selection technique whose goal is to find linear transformation that can reduce the data dimension while optimizing performance of classification on given data. (A preliminary short introduction of SFA appeared in our previous paper [6].) More specifically, let $A \in \Re^{n\times d}$ 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, considered as a column vector of size n, the vector of coefficients is given by $\alpha(I, A) = A^T I \in \Re^d$ and represents the orthogonal projection of I onto the subspace $\mathbb{S}_A$ spanned by the columns of A. Suppose the training data consists of representatives of C classes of images, with each class represented by k_{train} training images (denoted by $I_{c,1}, \ldots, I_{c,k_{train}}$) and k_{cross} cross validation images (denoted by $I'_{c,1}, \ldots, I'_{c,k_{cross}}$), the performance function F is defined as follows:

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

where,

$$\rho(I'_{c,i}, A) = \frac{\min_{c'\neq c,j} D^p(I'_{c,i}, I_{c',j}; A)}{\min_j D^p(I'_{c,i}, I_{c,j}; A) + \epsilon}\,. \qquad (2)$$

Here $p > 0$ is an exponent that can be adjusted to regularize p in different ways. The function ρ was used in the development of OCA with p=1. A large value $\rho(I'_{c,i}, A)$ indicates that the transformation A places $I_{c,i}$ lies much closer to a training sample of the class it belongs than to those of other classes; $\rho(I'_{c,i}, A) \approx 1$ indicates a transition between correct and incorrect decisions by the nearest neighbor classifier.

Scaling an entire ensemble does not change decisions based on KNN classifier. This is reflected in the fact that F is nearly scale invariant, that is $F(A) \approx F(\alpha A)$, for any $\alpha >$0. The function F only fails to be scale invariant due to the presence of ϵ in the denominator of Eq.(2), which is negligible in practice. Thus, one may restrict F to transformations of fixed norms (say, $\|A\| = 1$) without incurring any significant losses. Let

$$\mathbb{S} = \{A \in \Re^{d\times n} : \|A\| = tr(AA^T) = 1\}$$

be the unit sphere in $\Re^{d\times n}$. The goal of SFA is to maximize the performance function F over $\mathbb{S}$. In other words, to find

$$\hat{A} = \arg\max_{A\in\mathbb{S}} F(A) \qquad (3)$$

The linear mapping A: $\Re^n \to \Re^d$ is to be viewed as defined features of dimension $\leq d$, which are optimal from the standpoint of decisions made by the nearest neighbor classifier applied to the training set. Note that the existence of F is assumed by the simple facts that the sphere $\mathbb{S}$ is a compact space and F is continuous. This is in contrast with neighborhood component analysis, where no such theoretical assurance can be offered since the corresponding performance function is sensitive to scale.

3 Gradient Search Methods

For the OCA algorithm, the computational approach for estimation $\hat{A}$ is based on simulated annealing and the optimization process is carried out over a Grassmann manifold. This leads to heavy computational load as its sophicated geometry. However, in SFA, the optimization process is carried out over a sphere whose geometry is much simpler, thus it has significant computational advantage over OCA. A necessary condition for $\hat{A}$ is that, for any tangent vector at A, the direction derivative of F in the direction of the vector should be zero. Given $A \in \mathbb{S}$, to estimate the gradient vector field $\nabla_{\mathbb{S}}F$ on $\mathbb{S}$ associated with the performance function F, we first calculate $\nabla_{\Re^{d\times n}}F(A)$, the gradient of F viewed as a function on $\Re^{d\times n}$. Since F is nearly scale invariant,

$$\nabla_{\Re^{d\times n}}F(A) \approx \nabla_{\mathbb{S}}F(A) \qquad (4)$$

as the component of $\nabla_{\Re^{d\times n}}F(A)$ normal to the sphere is almost negligible. The numerical estimation of the left-hand side of (4) only involves standard procedures. Let E_{ij} be an $d \times n$ matrix such that, for $1 \leq i \leq d$, $1 \leq j \leq n$,

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

The partial derivative of F in the direction E_{ij} is estimated as

$$\partial_{ij}F(A) \approx \frac{F(A + \delta E_{ij}) - F(A)}{\delta} \qquad (5)$$

with $\delta > 0$ is small. Then, the gradient can be approximated by

$$\nabla_{\Re^{d\times n}}F(A) = \sum_{ij} \partial_{ij}F(A)E_{ij} \qquad (6)$$

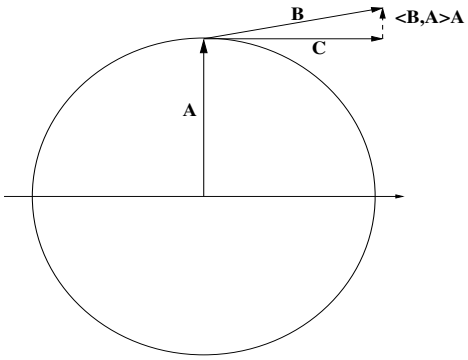

Figure 1. Estimation of $\nabla_{\mathbb{S}}F(A)$. Here $B = \nabla_{\Re^{d\times n}}F(A)$ and $C = \nabla_{\mathbb{S}}F(A)$.

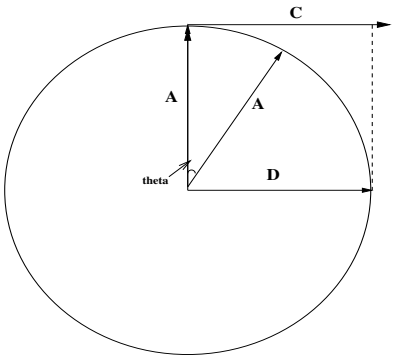

Figure 2. Geodesic updating of A. Here $C = \nabla_{\mathbb{S}}F(A)$ and $D = \frac{\nabla_{\mathbb{S}}F(A)}{\|\nabla_{\mathbb{S}}F(A)\|}$.

The gradient vector $\nabla_{\Re^{d\times n}}F(A)$ is nearly tangential to $\mathbb{S}$ at A; we enforce full tangentiality and obtain a more accurate estimation of $\nabla_{\mathbb{S}}F(A)$ by subtracting components normal to the sphere $\mathbb{S}$, as follows. For any $A \in \mathbb{S}$, a unit normal to $\mathbb{S}$ at A in $\Re^{d\times n}$ is given by A itself viewed as a vector in $\Re^{d\times n}$. Thus, we adopt the estimate

$$\nabla_{\mathbb{S}}F(A) \approx \nabla_{\Re^{d\times n}}F(A) - \langle \nabla_{\Re^{d\times n}}F(A), A\rangle A \qquad (7)$$

The estimation of $\nabla_{\mathbb{S}}F(A)$ is illustrated in Fig.1. We label $\nabla_{\Re^{d\times n}}F(A)$ as B in the figure. Obviously, the components normal to the sphere $\mathbb{S}$ can be computed as $< B, A > A$, where $< B, A >$ gives the vector length and A gives the direction of the normal. The gradient vector $\nabla_{\mathbb{S}}F(A)$, which labeled by C in the figure can be computed as $C = B - < B, A > A$, thus comes to Eq.(7).

Once we computed $\nabla_{\mathbb{S}}F(A)$, it is easy to get the update function for A. As shown in Fig.2, given a small move δ on the sphere, the angle between the old A and new A is $\theta = \delta\|\nabla_{\mathbb{S}}F(A)\|$, the update function for A is expressed as:

$$A = A\cos(\theta) + D\sin(\theta)$$

where $D = \frac{\nabla_{\mathbb{S}}F(A)}{\|\nabla_{\mathbb{S}}F(A)\|}$, thus,

$$A = A\cos(\delta\|\nabla_{\mathbb{S}}F(A)\|) + \frac{\nabla_{\mathbb{S}}F(A)}{\|\nabla_{\mathbb{S}}F(A)\|}\sin(\delta\|\nabla_{\mathbb{S}}F(A)\|) \qquad (8)$$

The deterministic gradient search algorithm is shown in Algorithm 1. Here are some remarks about Algorithm 1:

- The geodesic update of A described in step 4 of Algorithm 1 has the effect of displacing A by $\delta\|\nabla_{\mathbb{S}}F(A)\|$ units of length along the great circle of $\mathbb{S}$ through A in the direction $\nabla_{\mathbb{S}}F(A)$.

- After centering the data, we often initialize the search with the coordinate map $A : \Re^n \rightarrow \Re^d$ associated with the first d principal components of the training set. More precisely, let $v_1, \ldots, v_d$ be an orthonormal set of eigenvectors associated with the d dominant principal components of the covariance matrix of the training set, then, the search is initialized with the linear map $A(x) = (x \cdot v_1, \ldots, x \cdot v_d)^T$, where T denotes transposition.

Algorithm 1: Deterministic Gradient Search

1. Choose a threshold value $\epsilon > 0$ for the norm of the gradient and a step size $\delta > 0$.
2. Initialize the search with some $A \in \mathbb{S}$
3. Calculate $\nabla_{\mathbb{S}}F(A_t)$ using Eq.(1) and Eq.(7).
4. If $\|\nabla_{\mathbb{S}}F(A_t)\| < \epsilon$, set $\hat{A} = A$ and stop. Else, update A according to Eq.(8).
5. Go to step 3.

To deal with the case when A converges to a local maximum of $F(A)$, we add a stochastic component to A to achieve a global maximum solution. We call this as stochastic gradient search algorithm where a stochastic component is added to the deterministic gradient field $\nabla_{\mathbb{S}}F$ on $\mathbb{S}$. We skip the detail of the stochastic method in the paper, interested reader can refers to [6].

4 Two-stage SFA

Compared with OCA, which search the optimal basis on a Grassmann manifold, the computational cost of SFA is

much less as the search is on a sphere, a geometry much simpler than Grassmann manifold. However, the heavy search time is still a burden to its widely use in real applications. A significant reduction of the computational complexity can be achieved by restricting the SFA search to d dimensional subspaces of the span of the training images. If the dataset contains N images, $I_1, \ldots, I_N$, we arrange them as $D_N = [I_1, I_2, \ldots, I_N] \in \Re^{n \times N}$. If the rank of D_N is r, let D be an $n \times r$ matrix such $D^T D = I_r$ and whose columns form a basis of the span of the training set. Then, $D^T I \in \Re^r$ gives a reduced representation of an image $I \in \Re^n$. In typical recognition problems based on images, $r \ll n$, so that the SFA search can be carried out much more efficiently in this r-dimensional representation as $A \in \Re^{d \times r}$ space instead of $\Re^{d \times n}$. Note that, in this type of preliminary dimension reduction, all the information contained in the original training set is retained.

This gives rise to a two-stage SFA algorithm. Instead of solving the SFA optimization in the original image space, we limit the search to the span of the training images using a lower dimensional representation. To achieve even higher efficiency, in practice, we may want to further reduce the dimension using a computationally efficient dimension reduction method first. We refer to this step as pre-dimension reduction. An immediate question is how to choose pre-dimension reduction technique. Note that the performance is essentially determined by the distance between images in the reduced space, therefore, any method that retains the effective discriminative subspace would be sufficient. Two choices seem to be most relevant. First, we can choose to minimize the average reconstruction error, which can be achieved using PCA. An alternative is to choose the components that are most discriminative assuming the underlying distributions are Gaussian with fixed variance; this can be achieved using LDA by solving a generalized eigenvalue problem. However, as pointed out earlier, there is no theoretical basis for choosing PCA or LDA, in general.

To summarize, our two-stage SFA method is implemented as follows: in the first stage, we reduce the input data from the original high dimension to a lower dimension using a computationally efficient method; in the second stage, an SFA search is performed in the reduced space. As the search space is (much) smaller than the original one, the computational cost is greatly reduced.

5 Experimental Results

We present a set of experiments to evaluate the recognition accuracy and efficiency of SFA and two-stage SFA algorithms. The classification accuracy is measured by a 1-nearest neighbor(1NN) classifier. Program is run on a workstation with an Intel Xeon 3.00GHz CPU and 8.0G RAM.

We first evaluate the performance of SFA on the balance, ionosphere, iris, wine and housing data in UCI Machine learning Repository [2]. We split each data set into training (70%) and test (30%) subsets. Figure 3 shows the comparative training and more importantly testing performance of PCA, LDA, NCA and SFA on these data set. From this figure, we can see that except the training performance on ionosphere dataset, SFA is consistently better than other methods.

We also have evaluated the performance of SFA on the ORL [9] and AR face dataset. ORL face data set contains 400 face images of 40 individuals. The face images are perfectly centralized and the image size is $92 \times 112 = 10,304$. AR face data set [8] is a large face image data set and the recognition is more difficult than ORL. The instance of each face may contain significantly large areas of occlusion, due to the presence of sun glasses and scarves. The existence of occlusion dramatically increase the within-class variations of AR face image data. In this study, we use a subset of AR containing 1,638 face images of 126 individuals. Its image size is $768 \times 576 = 8,888$.

As the dimension of the original face image is high, the two-stage SFA instead of the original SFA is applied in the face recognition experiments to speed up the optimal basis search. For both of the dataset, PCA is used to reduced the dimension of original data to 100 in the first stage. We run the search for 1000 iteration and the subspace dimension d is set to 10. The classification accuracy is measured by a 1-nearest neighbor(1NN) classifier using 10-fold cross validation.

Unlike other stochastic optimal basis searching method, such as OCA, in which the optimal basis is obtained by searching on a Grassmann manifold, SFA shows its efficiency as the search process is performed on a sphere. In order to illustrate the efficient of SFA search, we compared the recognition performance and running time of SFA with OCA. Table 1 shows the running time and classification accuracy of SFA and OCA on ORL and AR data set, with different dimension case. The OCA cost time and accuracy are shown in the parenthesis. Note that when the dimension is in the original dimension space (10304 for ORL and 8888 for AR), the original OCA and SFA algorithms is used, while in the reduced dimension spaces, the two-stage OCA and two-stage SFA is used. We can see that compared with OCA, the searching time of SFA is largely reduced while the classification accuracy is still comparable.

Fig.4 shows the evolution of performance ratio F and recognition accuracy of the SFA algorithm on ORL and AR data sets. The left figures in each row show the evolution of performance ratio F, we can see for the training data, F in generally increased while it is not the truth for test data set. This is easy to understand, note that we are aim to optimize the performance ratio on the train data set, thus the optimal basis we obtained in each iteration is subject to op-

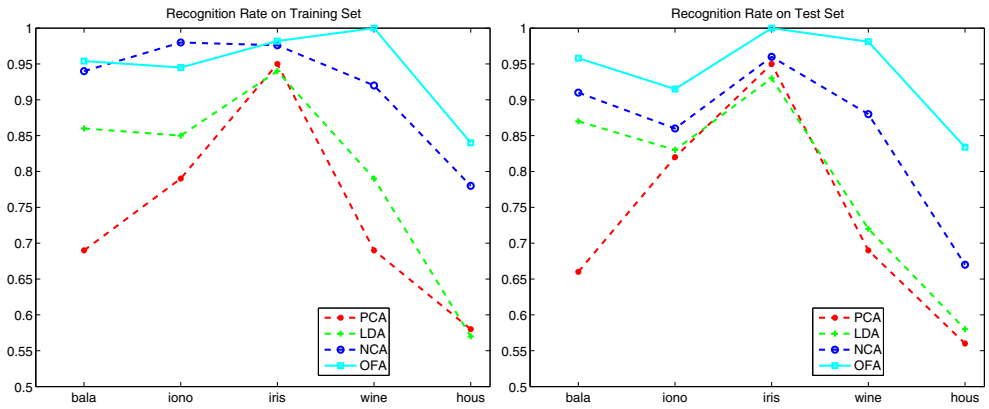

Figure 3. Evolution of recognition accuracy on UCI data sets.

Table 1. computational complexity (measured in seconds per iteration) and classification accuracy (%) comparison of SFA and OCA

ORL	dimension	10304	199	100	50	20
	Time	48.3(173)	1.23(10.00)	0.62(2.10)	0.32(0.97)	0.13(0.67)
	Accuracy	100(100)	100(100)	100(100)	100(100)	100(100)
AR	dimension	8888	500	300	50	20
	Time	235(421)	6.97(15.20)	4.23(8.43)	0.87(1.07)	0.38(0.54)
	Accuracy	96.03(92.12)	96.29(97.01)	96.03(93.66)	94.11(93.66)	92.86(93.02)

timize the training set images classification. Take note that in Fig.4 (c), there are some decrease of the ratio F in for AR training set, especially after iteration of 700. This is caused by the accept-reject criterion of the Metropolis-Hastinig algorithm. The right figures in each row show corresponding recognition accuracy for these two data sets. We can see the recognition accuracy in generally increased for both training and test data sets. Furthermore, the better performance ratio does not necessary generate the better recognition accuracy. Take the training set of (c) and (d) for an example, when the performance ratio is sometimes decreased after iteration 700, the corresponding classification accuracy is still increase. The reason is we define the optimal function not directly related to the classification accuracy. However, as the figure shows, an improvement of F will gradually lead to the improvement of recognition accuracy.

6 Summary

In this paper, we have provide a detail discussion of one linear representation method called SFA. SFA is developed for simultaneous dimension reduction and optimal feature selection for data parcellation and labeling problem based on the KNN classifier. The goal of SFA is to find a linear transformation $A : \Re^n \rightarrow \Re^d$ that optimizes the discriminative ability of the KNN classifier on the data. One seeks a linear mapping A that transforms each class into a cluster that is as compact as possible relative to the separation of the various clusters. The optimal basis A is obtained by a gradient search method over a unit sphere to obtained the optimal basis that can maximum of performance function F.

Although the search process of SFA is much simpler than OCA, by the nature of the stochastic optimization, the computational cost of SFA is still heavy. Based on this, we propose a two stage SFA method which can further reduce the its computational cost. In the first stage, we project the original data into a lower dimension subspace using some traditional dimension reduction methods, such as PCA and LDA. The stochastic search of SFA is on the second stage. As the search space is much simpler than the original sphere, the computational cost will be reduced greatly. We have tested the recognition performance of SFA and two stage SFA on several data sets and shows good performance on these data sets.

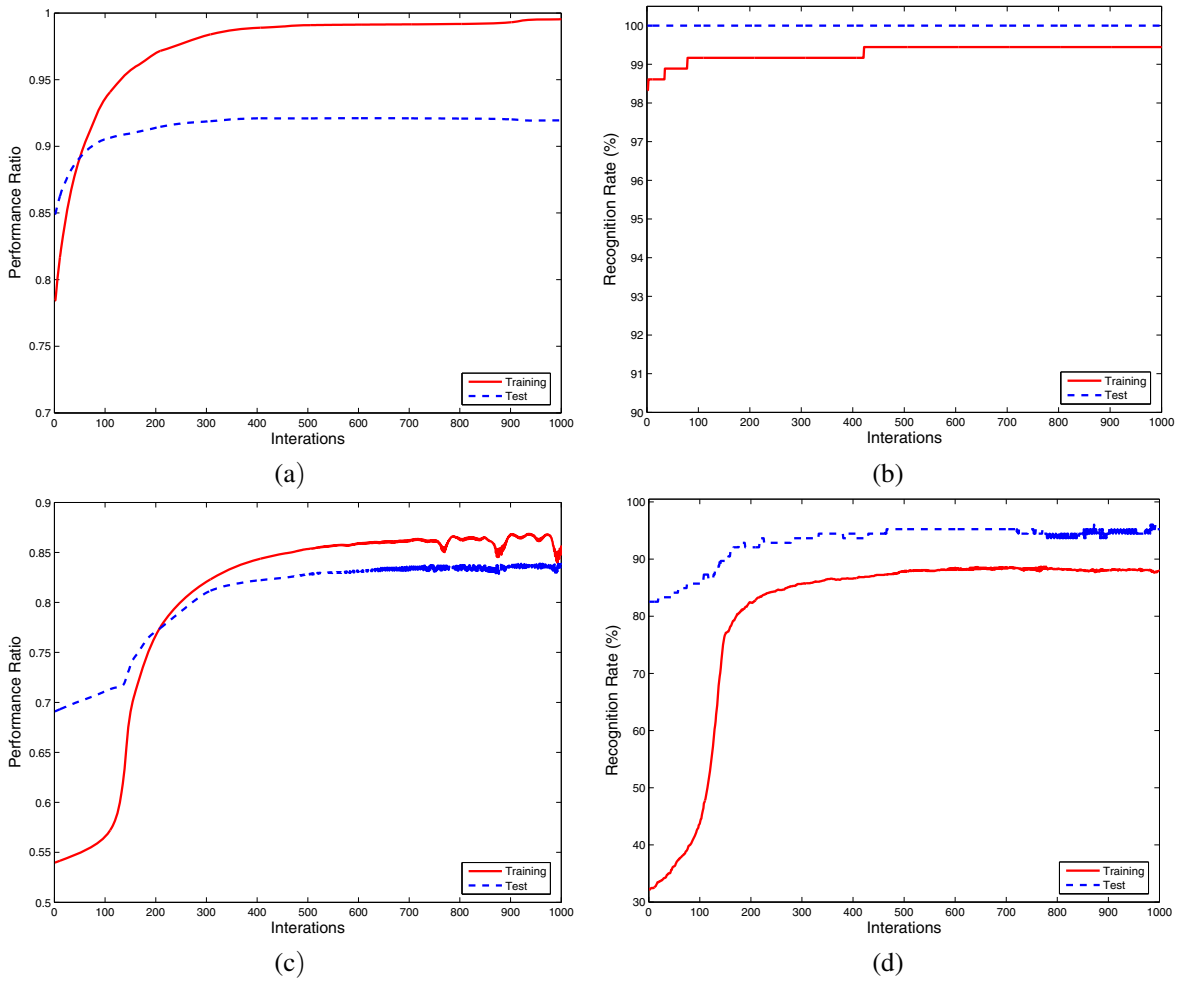

Figure 4. Evolution of performance ratio and classification accuracy on ORL and ARL data sets. (a): performance ratio of ORL; (b): classification accuracy of ORL. (c): performance ratio of AR; (d): classification accuracy of AR

7 Acknowledgments

This research was supported by NSF grant CCF-0514743 and DMS-0713012, and the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 RR021813.

References

[1] E. O. A. Hyvarinen, J. Karhunen. *Indenpendent Component Analysis*. John Wiley and Son, 2001.

[2] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.

[3] R. O. Duda, P. E. Hart, and D. Stock. *Pattern Classification*. Wiley, 2000.

[4] J. Goldberger, S. Roweis, G. Hinton, and R. Salakhutdinov. Neighborhood component analysis. *Advance in Neural Information and Processing System*, pages 513–520, 2005.

[5] I. T. Jolliffe. *Principal Componment Analysis*. Springer-Verlag, 1986.

[6] X. Liu and W. Mio. Splitting factor analysis and multi-class boosting. In *Proc. 13th International Conference on Image Processing (ICIP)*, Atlanta, GA, 2006.

[7] X. Liu, A. Srivastava, and K. Gallivan. Optimal linear representation of images for object recognition. *IEEE Transacation on Pattern Analysis and Machine Intelligence*, 26(5):662–666, 2004.

[8] A. M. Martinez and R. Benavente. The AR face database. Technical report, CVC Technical Report number 24, June 1998.

[9] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification. In *Proceedings of 2nd IEEE Workshop on Applications of Computer Vision*, 1994.