

A Globally Convergent MCA Algorithm by Generalized Eigen-Decomposition

Jianbin Gao^{1,2,3}, Mao Ye^{1,2}*, Jianping Li¹, Qi Xia¹

¹ School of Computer Science & Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China

² State Key Lab for Novel Software Technology, Nanjing University, China

³ Penn Image Computing and Science Laboratory (PICSL), Dept. of Radiology, University of Pennsylvania, Philadelphia, PA, USA

E-mail: gao.jb@hotmail.com

Received 2 January 2011

Accepted 12 November 2011

Abstract

Minor component analysis (MCA) are used in many applications such as curve and surface fitting, robust beamforming, and blind signal separation. Based on the generalized eigen-decomposition, we present a completely different approach that leads to derive a novel MCA algorithm. First, in the sense of generalized eigen-decomposition, by using gradient ascent approach, we derive an algorithm for extracting the first minor eigenvector. Then, the algorithm used to extract multiple minor eigenvectors is proposed by using the orthogonality property. The proofs and rigorous theoretical analysis show that our proposed algorithm is convergent to their corresponding minor eigenvectors. We identify three important characteristics of these algorithms. The first is that the algorithm for extracting minor eigenvectors can be extended to generalized minor eigenvectors easily. The second is that the corresponding eigenvalues can be computed simultaneously as a byproduct of this algorithm. The third is that the algorithm is globally convergent. The simulations have been conducted for illustration of the efficiency and effectiveness of our algorithm.

Keywords: generalized eigen-decomposition, minor component analysis, eigenvector, eigenvalue

1. Introduction

Minor component analysis (MCA) is a statistical method for extracting eigenvectors associated with the smallest eigenvalue of covariance matrix of input data. There are many applications for MCA, such as curve and surface fitting¹, beamforming^{2,3}, adaptive signal processing^{4,5}, etc. Neural networks method for MCA has been studied in recent years. For these algorithms and their references, refer to 1,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21.

The learning algorithm of MCA neural networks is described by a stochastic discrete-time (SDT) system. Many authors have proven the convergence by using continuous ordinary differential equations (ODE) approximation²². Since the convergence of continuous ODE does not imply the convergence of corresponding discrete ones when the learning rate is a constant^{23,24}, we can not use continuous ODE approximation to prove the convergence. On the other hand, because the behavior of conditional expectation of the weight vector can be studied by de-

*Corresponding author.

terministic discrete-time(DDT) system, it is reasonable to study the SDT algorithm by its DDT system indirectly^{24,29}. The DDT system at least can partially illustrate the phenomena of the corresponding SDT system.

Currently, the studies of the convergence of the DDT system of PCA or MCA algorithms are very mature, see our recent works^{25,26,27,28,29}. For constant learning rates, the convergence problem of the MCA learning algorithms has been studied systematically recently via their DDT systems, see^{7,12,23,26,29,32,33,34,35}. Based on the careful analysis of convergence properties of the existing algorithms, many novel convergent algorithms have been proposed, for example, see^{25,26,27,32,33,34,35}. However, most of these algorithms are derived by using mathematical inductions for convergence purpose, which are not meaningful optimization functions.

In this paper, a novel algorithm is proposed based on a viewpoint of generalized eigen-decomposition(GED). The GED is to extract the generalized eigenvectors of a matrix pencil (A,B) , which is given as follows³⁶:

$$Aw = \lambda Bw \tag{1}$$

where A and B are $n \times n$ positive definite and symmetric matrices in most signal processing applications. The positive scalar λ and the corresponding vector w are called generalized eigenvalue and generalized eigenvector of matrix pencil (A,B) respectively. Without loss of generality, assume the matrix pencil has n positive generalized eigenvalues, $\lambda_1 > \dots > \lambda_n$, and their corresponding generalized eigenvectors ϕ_1, \dots, ϕ_n , the following properties will hold³⁶:

$$A\phi_i = \lambda_i B\phi_i \tag{2}$$

$$\phi_i^T A \phi_j = \delta_{ij}, i, j \in \{1, \dots, n\} \tag{3}$$

where δ_{ij} is the Kronecker delta function. For the case of some eigenvalue with multiplicity larger than one, the problem is more difficult. Our algorithm cannot guarantee the convergence. We refer to ϕ_1 and ϕ_n as the first principal generalized eigenvector and the first minor generalized eigenvector respectively. In many practical applications such as

dimension reduction and signal processing, extracting the principal or minor generalized eigenvectors adaptively are critical^{30,31}.

Noted that $B\phi_i = \frac{1}{\lambda_i}A\phi_i$, let $\tilde{\phi}_i = \frac{1}{\sqrt{\lambda_i}}\phi_i$, we have that

$$B\tilde{\phi}_i = \frac{1}{\lambda_i}A\tilde{\phi}_i, \tag{4}$$

$$\tilde{\phi}_i^T A \tilde{\phi}_j = \delta_{ij}. \tag{5}$$

It can be easily shown that $\tilde{\phi}_n$ is the first principal generalized eigenvector of matrix pencil (B,A) . If $\tilde{\phi}_n$ is computed, the first minor generalized eigenvector (A,B) can be obtained accordingly as following,

$$\phi_n = \frac{\tilde{\phi}_n}{\sqrt{\tilde{\phi}_n^T B \tilde{\phi}_n}}. \tag{6}$$

Let $B = I_n$, to extract the minor eigenvector of matrix A is equivalent to find the minor generalized eigenvector of matrix pencil (A,B) . So we can extract the minor eigenvector of matrix A by finding the principal generalized eigenvector of matrix pencil (B,A) . Based on the above observations, we propose a new MCA algorithm from this completely different perspective. The major contribution of our paper is that by using a generalized eigen-decomposition approach MCA algorithms have three important characteristics. The first is that the algorithms for extracting minor eigenvectors and minor generalized eigenvectors are unified. The simulation results on extracting the minor generalized eigenvectors are shown in³⁷. The second is that the corresponding eigenvalues can be computed simultaneously as a byproduct of these algorithms. The third is that this algorithm is globally convergent. The rest of this paper is organized as follows. In section 2, the algorithm for extracting minor eigenvectors is derived from the perspective of generalized eigen-decomposition. The convergence analysis is presented in section 3. Section 4 provides the simulations in order to confirm the efficiency and effectiveness of our algorithms. In section 5, conclusions are presented.

2. The Algorithm

Consider a linear neuron with input $x(k) = [x_1(k), x_2(k), \dots, x_n(k)]^T \in R^n$, the weight vector $w(k) \in R^n$ and output $y(k) = w^T(k)x(k)$ at time k . $x(k) (k = 0, 1, \dots)$ is a zero-mean discrete-time stochastic process. Such process is constructed as a sequence $x(0), x(1), \dots$ of independent and identically distributed samples upon a distribution of a random variable. And weight vector $w(k)$ determines the relationship between input $x(k)$ and output $y(k)$ at time k .

Let covariance matrix $A = E\{x(k)x^T(k)\}$. Obviously the eigenvalues of this matrix are nonnegative. In our algorithm, some of eigenvalues can have a multiplicity more than one. However, for the simplicity of theoretical analysis, we assume $\lambda_1, \lambda_2, \dots, \lambda_n$ to be eigenvalues of A ordered by $\lambda_1 > \lambda_2 > \dots > \lambda_n > 0$. Since A is a symmetric matrix, then there exists an orthonormal basis of R^n composed by eigenvectors of matrix A . Suppose that $\{\phi_i | i = 1, \dots, n\}$ is an orthonormal basis in R^n and each ϕ_i is a unit eigenvector of A corresponding to the eigenvalue λ_i .

As stated before, to extract the minor eigenvectors of matrix A is equivalent to find the principal generalized eigenvectors of matrix pencil (B, A) where $B = I_n$. Based on this observation, we can derive our new algorithm directly by using an objective function. Assume w_1, \dots, w_{j-1} have already converged to the $j-1$ principal generalized eigenvectors of matrix pencil (B, A) , taking into account the constrains $w_i^T A w_j = \delta_{ij}$ for $i = 1, \dots, j-1$, the objective function to extract the j th principal generalized eigenvector of matrix pencil (B, A) is

$$J(w_j) = w_j^T B w_j + \sum_{i=1}^{j-1} \alpha_i w_j^T A w_i + \mu (w_j^T A w_j - 1),$$

where μ and α_i are Lagrange multipliers. When the function $J(w_j)$ reaches the maximum value, the principal generalized eigenvector of matrix pencil (B, A) will be the maximum solution of $J(w_j)$. The optimal value of μ and α_i can be determined by multiplying the gradient of $J(w_j)$ with respect to w_j by w_j^T or w_i^T from the left, and equating the result to zero. Taking into account that $w_i^T A w_j = \delta_{ij}$,

the optimum value μ and α_i are $\mu = -w_j^T B w_j$, $\alpha_i = -2w_i^T B w_j$.

Substituting these optimum value to J , the gradient of $J(w_j)$ with respect to w_j is

$$\begin{aligned} \nabla_{w_j} J(w_j) &= 4Bw_j - 2Bw_j w_j^T A w_j \\ &\quad - 2 \sum_{i=1}^{j-1} (w_j^T A w_i B w_i + w_j^T B w_i A w_i) \\ &\quad - 2w_j^T B w_j A w_j. \end{aligned}$$

By using gradient ascend method, let $W(k) = [w_1(k), \dots, w_j(k)]$ and $\|W(k)\|^2 = [\|w_1(k)\|^2, \dots, \|w_j(k)\|^2]$, we obtain the following algorithm in matrix form,

$$\begin{aligned} W(k+1) &= W(k) + \eta (2BW(k) \\ &\quad - AW(k)UT(W(k)^T BW(k)) \\ &\quad - BW(k)UT(W(k)^T AW(k)) \\ &\quad \cdot \emptyset \|W(k)\|^2, \end{aligned} \quad (7)$$

where η is the learning rate and $UT[\cdot]$ sets all elements below the diagonal of its matrix argument to zero, i.e. upper triangular. And $W(k)\emptyset \|W(k)\|^2 = [w_1(k)/\|w_1(k)\|^2, \dots, w_j(k)/\|w_j(k)\|^2]$. Similarly, assume $y_i(k) = w_i(k)^T x(k)$, considering $B = I_n$, the SDT version of algorithm is

$$\begin{aligned} w_j(k+1) &= w_j(k) + \eta (2w_j(k) \\ &\quad - \sum_{i=1}^j x(k)y_i(k)w_j(k)^T w_i(k) \\ &\quad - \sum_{i=1}^j w_i(k)y_j(k)y_i(k))/\|w_j(k)\|^2. \end{aligned} \quad (8)$$

Remark 1. When the above algorithm converges, the weight vector is different with the corresponding eigenvector with a scale factor which equals to the inverse of square root of its associated eigenvalue. After the process of normalization, the weight vector will be our desired minor eigenvector.

3. Theoretical Analysis

Direct analysis of the stochastic discrete time learning algorithm (8) is very difficult. Taking the conditional expectation $E\{w_j(k+1)|w_j(0), x(i), i < k\}$

to this algorithm and identifying the conditional expected value as the next iteration, we have its corresponding DDT algorithm (7). Convergence analysis of this DDT algorithm can partially illustrate the convergence phenomena of its SDT algorithm. Here, we can only show the convergence of the algorithm for sequentially extracting minor eigenvectors.

Mathematic induction method is used. First, when $j = 1$, we will show the convergence of the following algorithm

$$w_1(k+1) = w_1(k) + \eta \left(2 \frac{w_1(k)}{\|w_1(k)\|^2} - Aw_1(k) - w_1(k) \frac{w_1^T(k)Aw_1(k)}{\|w_1(k)\|^2} \right). \tag{9}$$

To show the weight vector $w_1(k)$ converges to the minor eigenvector with a scalar factor, three steps are used. First, we show that the weight vector of algorithm is bounded in iteration, i.e., our algorithm is feasible in practice. Then, that the direction of weight vector converges to that of the eigenvector associated with the smallest eigenvalue will be proved. In the end, by using the results of previous two steps, we show that the norm of weight vector will converge to a constant, thus proving the convergence.

Since vector set $\{\phi_1, \dots, \phi_n\}$ forms an orthogonal basis of R^n , for each $k \geq 0$, $w_1(k)$ can be represented as

$$w_1(k) = z_n(k)\phi_n + \sum_{j=1}^{n-1} \varepsilon_j(k)\phi_j \tag{10}$$

where $z_n(k)$ and $\varepsilon_j(k)$ are constants. Substitute (10) into (9), it follows that

$$z_n(k+1) = \left[1 + \eta \left(\frac{2}{\|w_1(k)\|^2} - \lambda_n - \beta_1(k) \right) \right] z_n(k), \tag{11}$$

and

$$\varepsilon_j(k+1) = \left[1 + \eta \left(\frac{2}{\|w_1(k)\|^2} - \lambda_j - \beta_1(k) \right) \right] \varepsilon_j(k), \tag{12}$$

for $1 \leq j \leq n - 1$, where $k \geq 0$, and $\beta_1(k) \equiv \frac{w_1^T(k)Aw_1(k)}{\|w_1(k)\|^2}$. Clearly, $\lambda_n \leq \beta_1(k) \leq \lambda_1$ for $k \geq 0$.

Lemma 1. Suppose that there exists a constant $\eta_0 > 0$ such that

$$\eta_0 < \eta < \frac{1}{2\lambda_1}, \tag{13}$$

and $w_1(0)$ is not orthogonal to the eigen-subspace V_{λ_n} , then there exists a time k_1 such that $\|w_1(k)\|^2 \in [c_0, c_1]$ for $k \geq k_1$, where $c_0 = [1 - 2\eta\lambda_1]^2 \frac{1}{\lambda_1}$ and $c_1 = \left[1 + 2\eta \left(\frac{1}{c_0} - \lambda_n \right) \right]^2 \frac{1}{\lambda_n}$.

Proof: Given in the Appendix.

Remark 2. It is required that initial value $w_1(0)$ is not orthogonal to eigensubspace V_{λ_n} . If $w_1(0)$ is orthogonal to V_{λ_n} , i.e., $w_1(0) \in V_{\lambda_n}^\perp$, where $V_{\lambda_n}^\perp$ is the orthogonal complement space of V_{λ_n} , the convergence is not guaranteed. However, in practical applications, the unstable situation cannot be observed since the dimension of $V_{\lambda_n}^\perp$ is less than that of R^n , and any small disturbance can make $w_1(0)$ non-orthogonal to V_{λ_n} .

Lemma 1 shows that $\|w_1(k)\|$ will be bounded below and above if some initial conditions are satisfied, thus, algorithm (9) is reasonable.

Lemma 2. Under the conditions of Lemma 1, the angle between $w_1(k)$ of algorithm (9) and eigenvector ϕ_n will approach to zero. Moreover,

$$\beta_1(k) - \lambda_n \leq \sum_{j=1}^{n-1} (\lambda_j - \lambda_n) \cdot \theta^{k-k_1} \cdot \left[\frac{\varepsilon_j(k_1)}{z_n(k_1)} \right]^2, \tag{14}$$

for $k > 0$, and

$$\sum_{j \neq n} \left[\frac{\varepsilon_j(k+1)}{z_n(k+1)} \right]^2 \leq \theta^{k+1-k_1} \cdot \sum_{j \neq n} \left[\frac{\varepsilon_j(k_1)}{z_n(k_1)} \right]^2 \rightarrow 0$$

as $k \rightarrow +\infty$, where

$$\theta = \left[1 - \frac{\eta(\lambda_{n-1} - \lambda_n)}{1 + 2\eta/c_0} \right]^2 \tag{15}$$

for $k > k_1$.

Proof: Given in the Appendix.

Lemma 2 means that the direction of weight vector approaches to the direction of vector ϕ_n .

Lemma 3. Assume the conditions of Lemma 1 hold, and $2\eta(\lambda_1 + \lambda_n + \sqrt{\lambda_1\lambda_n}) < 1$, then there exist constants k_1, Π_1, θ_1 and θ_2 so that

$$|z_n(k+1) - \frac{1}{\sqrt{\lambda_n}}| \leq (k - k_1 + 1) \cdot \Pi_1 \cdot \left[e^{-\theta_2(k-k_1+1)} + \max \left\{ e^{-\theta_1(k-k_1)}, e^{-\theta_2(k-k_1)} \right\} \right]$$

for $k \geq k_1$, where $\theta_1 = -\ln\theta$ and $\theta_2 = -\ln\delta$ with $\delta = 1 - \frac{2\eta\sqrt{\lambda_n}}{\sqrt{c_0}} - 2\eta\lambda_n$. Clearly, $\theta_1, \theta_2 > 0$ and $\delta < 1$.

Proof: Given in the Appendix.

Theorem 1 If the conditions of Lemma 1 and Lemma 3 are satisfied, then the $w_1(k)$ of algorithm (9) will converge to the eigenvector ϕ_n with a scale factor $\frac{1}{\sqrt{\lambda_n}}$ as $k \rightarrow +\infty$.

Proof: By Lemma 3, clearly,

$$\lim_{k \rightarrow +\infty} z_n(k) = \frac{1}{\sqrt{\lambda_n}}. \quad (16)$$

Then, using (16), and Lemma 2,

$$\varepsilon_j^2(k) = \left[\frac{\varepsilon_j(k)}{z_n(k)} \right]^2 \cdot z_n^2(k) \rightarrow 0 \quad (17)$$

as $k \rightarrow +\infty$.

From (10) together with (16) and (17), it follows that

$$\lim_{k \rightarrow +\infty} w_1(k) = \frac{\phi_n}{\sqrt{\lambda_n}} \in V_{\lambda_n}.$$

The proof is completed.

Up to now, the convergence has been shown for the case of $j = 1$. Assume for $i \leq j - 1$, w_i have already converged to $\frac{1}{\sqrt{\lambda_{n-i+1}}}\phi_{n-i+1}$, we will prove the convergence of w_j in the remaining part of this

section. The DDT version of $w_j(k)$ is

$$w_j(k+1) = w_j(k) + \eta \left(2w_j(k) - w_j(k)w_j(k)^T Aw_j(k) - Aw_j(k) - \sum_{i=1}^{j-1} Aw_i(k)w_i(k)^T w_j(k) - \sum_{i=1}^{j-1} w_i(k)w_i(k)^T Aw_j(k) \right) \div \|w_j(k)\|^2. \quad (18)$$

Expressing $w_j(k)$ as that in (10), we have the following equations,

$$\alpha_i(k+1) = [1 - \eta(\lambda_i + \beta_j(k))] \alpha_i(k), \quad (19)$$

for $n - j + 1 < i \leq n$,

$$z_{n-j+1}(k+1) = \left[1 + \eta \left(\frac{2}{\|w_j(k)\|^2} - \lambda_{n-j+1} - \beta_j(k) \right) \right] z_{n-j+1}(k), \quad (20)$$

and

$$\varepsilon_i(k+1) = \left[1 + \eta \left(\frac{2}{\|w_j(k)\|^2} - \lambda_i - \beta_j(k) \right) \right] \varepsilon_i(k), \quad (21)$$

for $1 \leq i < n - j + 1$, where $k \geq 0$, and $\beta_j(k) \equiv \frac{w_j^T(k)Aw_j(k)}{\|w_j(k)\|^2}$. Clearly, $\lambda_n \leq \beta_j(k) \leq \lambda_1$ for $k \geq 0$.

Let $\Pi(k) = \sum_{i=1}^{n-j} \varepsilon_i(k)^2 + z_{n-j+1}(k)^2$ and $\Lambda(k) = \sum_{i=n-j+2}^n \alpha_i(k)^2$.

Lemma 4. With the same conditions in Lemma 1, then there exists time k_1 such that $\|w_j(k)\|^2 \in [c_0, c_1]$ for $k \geq k_1$ where c_1 is the same definition as that in Lemma 1, $c_0 = [1 - 2\eta\lambda_1]^2 (\frac{1}{\lambda_1} - c_2)$ and c_2 is a constant less than $\frac{1}{\lambda_1}$. Moreover, $\lim_{k \rightarrow \infty} \alpha_i(k) = 0$ exponentially for $n - j + 1 < i \leq n$.

Proof: Given in the Appendix.

By using the same method in Lemma 2, we have the following lemma,

Lemma 5. Under the conditions of Lemma 4, then the angle between $w_j(k)$ of algorithm (18) and

eigenvector ϕ_{n-j+1} will approach to zero. Moreover,

$$\begin{aligned} \left| \beta_j(k) - \lambda_{n-j+1} \right| &\leq \sum_{i=1}^{n-j} (\lambda_i - \lambda_{n-j+1}) \cdot \theta^{k-k_1} \\ &\quad \cdot \left[\frac{\varepsilon_i(k_1)}{z_{n-j+1}(k_1)} \right]^2 + \sum_{i=n-j+2}^n \left| \lambda_i \right. \\ &\quad \left. - \lambda_{n-j+1} \right| \cdot \delta_1^{k-k_1} \\ &\quad \cdot \left[\frac{\alpha_i(k_1)}{z_{n-j+1}(k_1)} \right]^2, \end{aligned}$$

for $k > 0$, and

$$\begin{aligned} \sum_{i=1}^{n-j} \left[\frac{\varepsilon_i(k+1)}{z_{n-j+1}(k+1)} \right]^2 &\leq \theta^{k+1-k_1} \\ &\quad \cdot \sum_{i=1}^{n-j} \left[\frac{\varepsilon_i(k_1)}{z_{n-j+1}(k_1)} \right]^2 \rightarrow 0 \end{aligned}$$

as $k \rightarrow +\infty$, where $\delta_1 = 1 - 2\eta\lambda_n$ and

$$\theta = \left[1 - \frac{\eta(\lambda_{n-j+2} - \lambda_{n-j+1})}{1 + 2\eta/c_0} \right]^2 \quad (22)$$

for $k > 0$.

Similarly, the following lemma holds.

Lemma 6. Assume the conditions of Lemma 4 hold, and $2\eta(\lambda_1 + \lambda_{n-j+1} + \sqrt{\lambda_1\lambda_{n-j+1}}) < 1$, there exist constants k_1, Π_1, θ_1 and θ_2 so that

$$\begin{aligned} \left| z_{n-j+1}(k+1) - \frac{1}{\sqrt{\lambda_{n-j+1}}} \right| &\leq (k - k_1 + 1) \cdot \Pi_1 \\ &\quad \cdot \left[e^{-\theta_2(k-k_1+1)} \right. \\ &\quad \left. + \max \left\{ e^{-\theta_1(k-k_1)}, \right. \right. \\ &\quad \left. \left. e^{-\theta_2(k-k_1)} \right\} \right] \end{aligned}$$

for $k \geq k_1$, where $\theta_1 = -\ln(\max\{\theta, \delta_1\})$ and $\theta_2 = -\ln\delta$ with $\delta = 1 - \frac{2\eta\sqrt{\lambda_{n-j+1}}}{\sqrt{c_0}} - 2\eta\lambda_{n-j+1}$. Clearly, $\theta_1, \theta_2 > 0$ and $\delta < 1$.

Theorem 2 If the conditions of Lemma 4 and Lemma 6 are satisfied, the $w_j(k)$ of algorithm (18) will converge to the eigenvector ϕ_{n-j+1} with a scale factor $\frac{1}{\sqrt{\lambda_{n-j+1}}}$ as $k \rightarrow +\infty$.

Proof: By Lemma 6, clearly,

$$\lim_{k \rightarrow +\infty} z_{n-j+1}(k) = \frac{1}{\sqrt{\lambda_{n-j+1}}}. \quad (23)$$

Then, using (23), and Lemma 4, 5,

$$\varepsilon_j^2(k) = \left[\frac{\varepsilon_j(k)}{z_{n-j+1}(k)} \right]^2 \cdot z_{n-j+1}^2(k) \rightarrow 0 \quad (24)$$

for $1 \leq j < n - j + 1$ and $\lim_{k \rightarrow \infty} \alpha_i^2(k) = 0$ exponentially for $n - j + 1 < i \leq n$.

From (10) together with (23) and (24), it follows that

$$\lim_{k \rightarrow +\infty} w_j(k) = \frac{\phi_{n-j+1}}{\sqrt{\lambda_{n-j+1}}} \in V_{\lambda_{n-j+1}}.$$

The proof is completed.

Remark 3. By Theorem 1 and Theorem 2, we know that our algorithm can not only extract the minor eigenvector, but also compute the corresponding eigenvalue whose value equals to $1/\|\tilde{w}\|^2$, where \tilde{w} represents a converged weight vector.

Remark 4. For choosing practical learning rate η to guarantee the convergence, practically, because the exact values of λ_1 and λ_n are unknown, since $Trace(A) = \sum_{i=1}^n \lambda_i$, we can roughly let $\eta = \frac{1}{3Trace(A)}$, where $Trace(A) = \sum_{k=1}^N x^2(k)/N$, which can be computed incrementally. Based on our analysis, the varying learning rate η does not affect the convergence. But theoretical analysis of the impact of the varying η is very difficult.

4. Simulations

In this section, we perform a few simulations on the algorithm (8) to confirm the convergence results and illustrate the good performance by using MATLAB software. There are many MCA algorithms, we cannot compare each of them. Because MÖLLER¹⁵ is a successfully improved MCA algorithm and OJA+¹⁶ is the original online MCA learning algorithm, we

only compare the performance of the novel algorithm with the performance of MÖLLER, and OJA+. For simplicity, the novel algorithm (8) is referred to as GM.

A sequence of 20-dimensional signals are generated, whose covariance matrix has the two smallest eigenvalues 0.3445, 0.3838 and the largest eigenvalue 1.9711. We generate the input sequence $x_k, 1 \leq k \leq 20000$ by repeating adding zero-mean gaussian noise to the previous signal sequence with 0.1 variance. If the number of samples is large enough, it is easy to show that the covariance matrix $A = E[x_k x_k^T]$ is the same as that of the signal covariance matrix. The following evaluation function is used to measure the accuracy of those algorithms. The direction cosine measures the accuracy of direction of the estimated eigenvectors,

$$\text{Direction Cosine}(k) = \frac{|w(k)^T \phi|}{\|w(k)\| \|\phi\|}, \quad (25)$$

where $w(k)$ is the weight vector at time k for any one of previous mentioned algorithms, ϕ is the actual corresponding eigenvector of matrix A.

In the first set of experiments, we compute the first minor eigenvector by using algorithms GM, OJA+ and MÖLLER. Let the learning rate $\eta = 0.01$ for all algorithms and the same initial weight vector is generated randomly with unit norm. The averaged value curves of evaluation function of 50 independent runs are drawn in Fig.1.

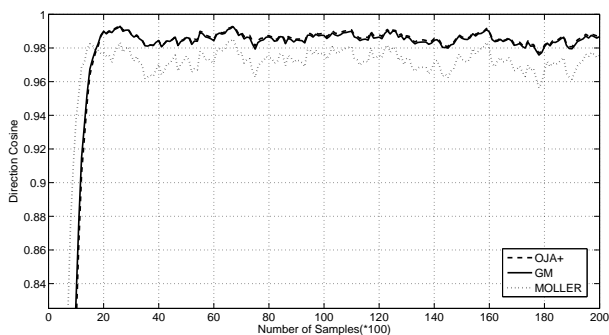


Fig. 1. Performance of extracting the first minor eigenvector using algorithms GM, OJA+ and MÖLLER respectively.

From Fig.1, we could observe that GM algorithm is convergent, which confirms the convergence theory. The rate of convergence of GM is almost the same as that of OJA+. And both of two algorithms are faster than the algorithm MÖLLER. Because

the OJA+ algorithm has the limitation that the minor eigenvalue must be less than 1, so our algorithm GM is more suitable in practical applications.

In the second set of experiments, we will show the impact of different choosing of learning rates. We still use the input sequence in the first set of experiments. To illustrate it, we perform algorithm GM to extract the first minor eigenvector with different learning rates $\eta = 0.02, 0.01, 0.005$ with the same initial weight vector. The value curves of evaluation function are drawn in Fig.2.

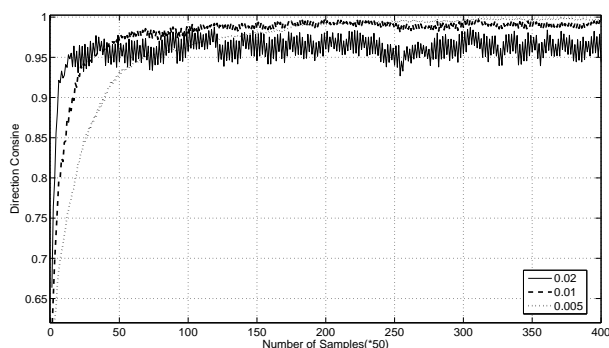


Fig. 2. Performance of algorithm GM with learning rate $\eta = 0.02, 0.01, 0.005$ respectively.

From Fig.2, it has been shown that the larger the learning rate is, the more oscillated the algorithm is. In the third set of experiments, the ability of algorithm GM to extract multiple minor eigenvectors is shown. Here, the same input sequence of previous experiments is used. The first three minor eigenvectors are computed using our algorithm GM. The learning rate $\eta = 0.01$. The value curves of evaluation function are drawn in Fig.3.

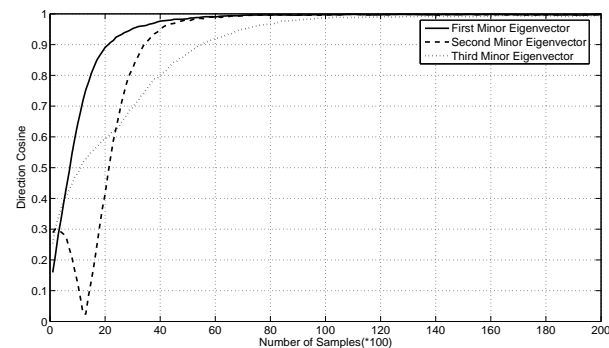


Fig. 3. Performance of algorithm GM for extracting the first three minor eigenvectors.

From Fig.3, the efficiency of algorithm GM is shown for extracting multiple minor eigenvectors.

Please note that the algorithms OJA+ and MöLLER can not extract multiple minor eigenvectors. Further experiments are performed to confirm the convergence of our algorithms with different standard deviations of the input data and more higher dimensional data, for the limitation of space, which are not shown here.

5. Conclusions

A completely new approach that leads to a novel algorithm for extracting minor eigenvectors online has been discussed. First, from the perspective of generalized eigen-decomposition the algorithm is proposed. Theoretical analysis shows that this algorithm is globally convergent to the minor eigenvectors. Simulations have been conducted for illustration of the convergence and efficiency of our algorithms. From both theoretical analysis and simulations, it can be concluded that our algorithm is a good choice in the practical applications.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (60702071), Program for New Century Excellent Talents in University (NCET-06-0811), 973 National Basic Research Program of China, (2010CB732501), Foundation of Sichuan Excellent Young Talents (09ZQ026-035) and Open Project of State Key Lab. for Novel Software Technology of Nanjing University.

Appendix

Proof of Lemma1:

Because $w(0)$ is not orthogonal to V_{λ_n} , we have $z_n(0) \neq 0$. If condition (13) is satisfied, it is easy to see that $\|w_1(k+1)\| \leq \|w_1(k)\|$ if $\|w_1(k)\|^2 > 1/\lambda_n$ and $\|w(k+1)\| \geq \|w_1(k)\|$ if $\|w_1(k)\|^2 < 1/\lambda_1$. So there exists a time k_1 such that $\|w_1(k_1)\|^2 \in [1/\lambda_1, 1/\lambda_n]$.

If condition (13) is satisfied, for each $j(1 \leq j \leq$

$n)$, we have

$$1 + \eta \left(\frac{2}{\|w_j(k)\|^2} - \lambda_j - \beta_1(k) \right) \geq 0 \quad (26)$$

for $k \geq 0$. From (26), we have that

$$\|w_1(k+1)\| \geq [1 + 2\eta (1/\|w_1(k)\|^2 - \lambda_1)] \cdot \|w(k)\|$$

and

$$\|w_1(k+1)\| \leq [1 + 2\eta (1/\|w_1(k)\|^2 - \lambda_n)] \cdot \|w_1(k)\|$$

for $\|w_1(k)\|^2 \in [c_0, c_1]$.

Next, we will show $\|w_1(k+1)\|^2 \in [c_0, c_1]$ if $\|w_1(k)\|^2 \in [c_0, c_1]$.

Case 1, if $1/\lambda_n \leq \|w_1(k)\|^2 \leq c_1$, it is easy to see that $\|w_1(k+1)\|^2 \leq \|w_1(k)\|^2 \leq c_1$ and $\|w_1(k+1)\|^2 \geq [1 + 2\eta (1/\|w_1(k)\|^2 - \lambda_1)]^2 / \lambda_n \geq [1 - 2\eta \lambda_1]^2 / \lambda_n > c_0$.

Case 2, for $\|w_1(k)\|^2 \in [1/\lambda_1, 1/\lambda_n]$, we have $\|w_1(k+1)\|^2 \leq [1 + 2\eta (1/\|w_1(k)\|^2 - \lambda_n)]^2 / \lambda_n \leq c_1$ and $\|w_1(k+1)\|^2 \geq [1 + 2\eta (1/\|w_1(k)\|^2 - \lambda_1)]^2 / \lambda_1 \geq c_0$.

Case 3, when $\|w_1(k)\|^2 \in [c_0, 1/\lambda_1]$, it follows that $\|w_1(k+1)\|^2 \geq \|w_1(k)\|^2 \geq c_0$ and

$$\begin{aligned} \|w_1(k+1)\|^2 &\leq \left[1 + 2\eta \left(\frac{1}{\|w_1(k)\|^2} - \lambda_n \right) \right]^2 \cdot \frac{1}{\lambda_1} \\ &\leq \left[1 + 2\eta \left(\frac{1}{c_0} - \lambda_n \right) \right]^2 \cdot \frac{1}{\lambda_1} \leq c_1. \end{aligned}$$

The proof is completed.

Proof of Lemma 2:

Clearly, $0 < \theta < 1$. It follows (11) and (12) that $z_n(k) \geq 0$ for all $k \geq 0$ if $z_n(0) \geq 0$, and $z_n(k) \leq 0$ for all $k \geq 0$ if $z_n(0) \leq 0$. Because $w_1(0)$ is not orthogonal to V_{λ_n} , $z_n(0) \neq 0$. Without loss of generality, we assume that $z_n(0) > 0$, and thus, $z_n(k) > 0$ for all $k \geq 0$.

From (11~12) and according to Lemma 1, it follows

that

$$\begin{aligned} & \left[\frac{\varepsilon_j(k+1)}{z_n(k+1)} \right]^2 \\ &= \left[\frac{1 + \eta(2/\|w_1(k)\|^2 - \lambda_j - \beta_1(k))}{1 + \eta(2/\|w_1(k)\|^2 - \lambda_n - \beta_1(k))} \right]^2 \cdot \left[\frac{\varepsilon_j(k)}{z_n(k)} \right]^2 \\ &= \left[1 - \frac{\eta(\lambda_j - \lambda_n)}{1 + \eta(2/\|w_1(k)\|^2 - \lambda_n - \beta_1(k))} \right]^2 \cdot \left[\frac{\varepsilon_j(k)}{z_n(k)} \right]^2 \\ &\leq \left[1 - \frac{\eta(\lambda_{n-1} - \lambda_n)}{1 + 2\eta/c_0} \right]^2 \cdot \left[\frac{\varepsilon_j(k)}{z_n(k)} \right]^2 \\ &\leq \theta^{k-k_1+1} \cdot \left[\frac{\varepsilon_j(k_1)}{z_n(k_1)} \right]^2 \rightarrow 0, \quad (j = 1, \dots, n-1) \end{aligned}$$

as $k \rightarrow +\infty$. This means that the angle between $w_1(k)$ and ϕ_n approaches to zero.

It follows that

$$\begin{aligned} \beta_1(k) - \lambda_n &= \frac{\sum_{j=1}^{n-1} (\lambda_j - \lambda_n) \varepsilon_j^2(k)}{z_n^2(k) + \sum_{j=1}^{n-1} \varepsilon_j^2(k)} \\ &\leq \sum_{j=1}^{n-1} (\lambda_j - \lambda_n) \cdot \theta^{k-k_1} \cdot \left[\frac{\varepsilon_j(k_1)}{z_n(k_1)} \right]^2 \end{aligned}$$

for $k > k_1$. The proof is completed.

Proof of Lemma 3:

From equation (11), it follows that

$$\begin{aligned} z_n(k+1) - \frac{1}{\sqrt{\lambda_n}} &= \left(z_n(k) - \frac{1}{\sqrt{\lambda_n}} \right) - \eta \left(\beta_1(k) \right. \\ &\quad \left. - \lambda_n \right) z_n(k) + 2\eta \left(\frac{1}{\|w_1(k)\|^2} \right. \\ &\quad \left. - \lambda_n \right) z_n(k) \end{aligned}$$

for $k > 0$. Since

$$\frac{1}{\lambda_n} - z_n(k)^2 = \left(\frac{1}{\sqrt{\lambda_n}} - z_n(k) \right) \left(\frac{1}{\sqrt{\lambda_n}} + z_n(k) \right),$$

for $k \geq k_1$, we have that

$$\begin{aligned} z_n(k+1) - \frac{1}{\sqrt{\lambda_n}} &\leq \left(z_n(k) - \frac{1}{\sqrt{\lambda_n}} \right) \cdot \left(1 - \right. \\ &\quad \left. \frac{2\eta\lambda_n}{z_n(k)} \left(\frac{1}{\sqrt{\lambda_n}} + z_n(k) \right) \right) + \gamma_1, \end{aligned}$$

where $\gamma_1 = |\eta(\beta_1(k) - \lambda_n)z_n(k)|$.

By conditions in Lemma 1, we have that

$$\begin{aligned} \Gamma &= 1 - \frac{2\eta\lambda_n}{z_n(k)} \left(\frac{1}{\sqrt{\lambda_n}} + z_n(k) \right) \\ &\geq 1 - \frac{2\eta\sqrt{\lambda_n}}{\sqrt{c_0}} \left(1 + \sum_{i \neq n} \frac{\varepsilon_i(k)^2}{z_n(k)^2} \right) - 2\eta\lambda_n. \end{aligned}$$

Combining above equality,

$$\begin{aligned} z_n(k+1) - \frac{1}{\sqrt{\lambda_n}} &\leq \left(z_n(k) - \frac{1}{\sqrt{\lambda_n}} \right) \left(1 - \frac{2\eta\sqrt{\lambda_n}}{\sqrt{c_0}} \right. \\ &\quad \left. - 2\eta\lambda_n \right) + \gamma_1 + \gamma_2, \end{aligned}$$

$$\text{where } \gamma_2 = -\frac{2\eta\sqrt{\lambda_n}}{c_0} \sum_{i \neq n} \frac{\varepsilon_i(k)^2}{z_n(k)^2} \left(z_n(k) - \frac{1}{\sqrt{\lambda_n}} \right).$$

Because $\|w(k)\|$ is bounded above and below for $k \geq k_1$, there are constants c_2, M_1, M_2 such that $|\gamma_1| \leq \eta c_2 |\beta_1 - \lambda_n| \leq M_1 \eta e^{-\theta_1(k-k_1)}$ and $|\gamma_2| \leq M_2 \eta e^{-\theta_1(k-k_1)}$. For $k \geq k_1$, by assumptions in Lemma 3, it follows that

$$\begin{aligned} \left| z_n(k+1) - \frac{1}{\sqrt{\lambda_n}} \right| &\leq \delta \left| z_n(k) - \frac{1}{\sqrt{\lambda_n}} \right| \\ &\quad + M \eta e^{-\theta_1(k-k_1)}, \end{aligned}$$

where M is a constant and $k \geq k_1$. Then, for $k > k_1$,

$$\begin{aligned} \left| z_n(k+1) - \frac{1}{\sqrt{\lambda_n}} \right| &\leq \delta^{k-k_1+1} \left| z_n(k_1) - \frac{1}{\sqrt{\lambda_n}} \right| \\ &\quad + \eta M \sum_{r=0}^{k-k_1} (\delta e^{\theta_1})^r e^{-\theta_1(k-k_1)} \\ &\leq \delta^{k-k_1+1} \left| z_n(k_1) - \frac{1}{\sqrt{\lambda_n}} \right| \\ &\quad + (k - k_1 + 1) \cdot \eta M \\ &\quad \cdot \max \left\{ \delta^{k-k_1}, e^{-\theta_1(k-k_1)} \right\} \\ &\leq (k - k_1 + 1) \cdot \Pi_1 \cdot \left[e^{-\theta_2(k-k_1+1)} \right. \\ &\quad \left. + \max \left\{ e^{-\theta_2(k-k_1)}, e^{-\theta_1(k-k_1)} \right\} \right] \end{aligned}$$

where Π_1 is a constant. The proof is completed.

Proof of Lemma 4:

Because $w_j(0)$ is not orthogonal to $V_{\lambda_{n-j+1}}$, we have $z_{n-j+1}(0) \neq 0$. If condition (13) is satisfied, it is easy to see that $\|w_j(k+1)\| \leq \|w_j(k)\|$ if $\|w_j(k)\|^2 > 1/\lambda_n$ and $\|w_j(k+1)\| \geq \|w_j(k)\|$ if $\|w_j(k)\|^2 < 1/\lambda_1$. So there exists a time k_1 such that $\|w_j(k_1)\|^2 \in [1/\lambda_1, 1/\lambda_n]$.

If condition (13) is satisfied, for each $i(1 \leq i \leq n)$ when $\|w_j(k)\|^2 \in [c_0, c_1]$, we have

$$1 + \eta (2/\|w_j(k)\|^2 - \lambda_i - \beta_j(k)) \geq 0. \quad (27)$$

From (27), we have that

$$\begin{aligned} \|w_j(k+1)\| &\geq \Pi(k+1) \\ &\geq [1 + 2\eta (1/\|w_1(k)\|^2 - \lambda_1)] \\ &\quad \cdot \Pi(k) \end{aligned}$$

and

$$\|w_j(k+1)\| \leq [1 + 2\eta (1/\|w_j(k)\|^2 - \lambda_n)] \cdot \|w_j(k)\|$$

for $k \geq k_1$.

Next, we will show $\|w_j(k+1)\|^2 \in [c_0, c_1]$ if $\|w_j(k)\|^2 \in [c_0, c_1]$. Case 1, if $1/\lambda_n \leq \|w_j(k)\|^2 \leq c_1$, it is easy to see that $\|w_j(k+1)\|^2 \leq \|w_j(k)\|^2 \leq c_1$ and $\|w_j(k+1)\|^2 \geq \Pi(k+1) \geq [1 - 2\eta\lambda_1]^2 \Pi(k) \geq [1 - 2\eta\lambda_1]^2 (1/\lambda_n - \Lambda(k))$.

Case 2, for $\|w_j(k)\|^2 \in [1/\lambda_1, 1/\lambda_n]$, we have $\|w_j(k+1)\|^2 \leq [1 + 2\eta (\lambda_1 - \lambda_n)]^2 / \lambda_n \leq c_1$ and $\|w_j(k+1)\|^2 \geq \Pi(k+1) \geq [1 + 2\eta (\lambda_n - \lambda_1)]^2 \cdot \Pi(k)$.

Case 3, when $\|w_j(k)\|^2 \in [c_0, 1/\lambda_1]$, it follows that $\|w_j(k+1)\|^2 \geq \|w_j(k)\|^2 \geq c_0$ and $\|w(k+1)\|^2 \leq [1 - 2\eta\lambda_1]^2 / \lambda_1 \leq c_1$.

If condition (13) is satisfied, by equation (19), it follows that $\Lambda(k+1) < \Lambda(k)$. So there exists time k_2 , such that $\Lambda(k) \leq c_2 < 1/\lambda_1$ for $k \geq k_2$. Without loss of generality, assume $k_2 \leq k_1$. Furthermore, it follows that in Case 1 and Case 2, $\|w_j(k+1)\|^2 \geq [1 - 2\eta\lambda_1]^2 (1/\lambda_1 - c_2) \geq c_0$.

Since $1 - \eta (\lambda_i + \beta_j(k)) \geq 1 - 2\eta\lambda_1 > 0$ for $k > 0$. Let $\delta_1 = 1 - 2\eta\lambda_n$, it follows that $|\alpha_i(k+1)| < \delta_1^{k+1} |\alpha_i(0)|$ for $k > 0$ and $n-j+1 < i \leq n$. It follows that $\lim_{k \rightarrow \infty} \alpha_i(k) = 0$ exponentially for $n-j+1 < i \leq n$. The proof is completed.

References

1. L. Xu, E. Oja and C. Suen, "Modified Hebbian learning for curve and surface fitting," *Neural Networks*, **5**, 441–457 (1992).
2. H. Cox, R. M. Zeskind and M. M. Owen, "Robust adaptive beamforming," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, **35**, 1365–1376 (1987).
3. S. Fiori and F. Piazza, "Neural MCA for robust beamforming," *Proc. IEEE Intl. Symp. on Circuits and Systems*, **III**, 614–617 (2000).
4. G. Mathew and V. Reddy, "Development and analysis of a neural network approach to Pisarenko's harmonic retrieval method," *IEEE Trans. on Signal Processing*, **42**, 663–667 (1994).
5. S. Ouyang, Z. Bao and G. Liao, "Adaptive step-size minor component extraction algorithm," *Electronics Letters*, **35**, 443–444 (1999).
6. T. Chen, S. Amari and Q. Lin, "A unified algorithm for principal and minor components extraction," *Neural Networks*, **11**, 385–390 (1998).
7. G. Cirrincione, M. Cirrincione, J. Heralut and S. Van Huffel, "The MCA EXIN neuron for the minor component analysis," *IEEE Trans. on Neural Networks*, **13**, 160–187 (2002).
8. K. I. Diamantaras and S. Y. Kung, *Principal component neural networks, theory and applications*, (Wiley, 1996).
9. T. Chen and S. Amari, "Unified Stabilization Approach to Principal and Minor Components Extraction Algorithms," *Neural Networks*, **14**, 1377–1387 (2001).
10. T. Chen, S. Amari and N. Murata, "Sequential Extraction of Minor Components," *Neural Processing Letters*, **13**, 195–201 (2001).
11. S. C. Douglas, S. -Y. Kong and S. Amari, "A self-stabilized minor subspace rule," *IEEE Signal Processing Letters*, **5**, 328–330 (1998).
12. D. Z. Feng, Z. Bao and L. C. Jiao, "Total least mean squares algorithm," *IEEE Trans. on Signal Processing*, **46**, 2122–2130 (1998).
13. F. Luo, R. Unbehauen and A. Cichocki, "A minor component analysis algorithm," *Neural Networks*, **10**, 291–297 (1997).
14. D. Peng and Z. Yi, "Dynamics of generalized PCA and MCA learning algorithms," *IEEE Trans. on Signal Processing*, **18**, 1777–1784 (2007).
15. R. MöLLER, "A self-stabilizing learning rule for minor component analysis," *Int. J. Neural Syst.*, **14**, 1–8 (2004).
16. E. Oja, "Principal components, Minor components, and linear neural networks," *Neural Networks*, **5**, 927–935 (1992).

17. E. Oja and L. Wang, "Robust fitting by non-linear neural units," *Neural Networks*, **9**, 435–444 (1996).
18. S. Ouyang, Z. Bao, G.-S. Liao and P. C. Ching, "Adaptive minor component extraction with modular structure," *IEEE Trans. on Signal Processing*, **49**, 2127–2137 (2001).
19. L. Wang and J. Karhunen, "A unified neural bigradient algorithm for robust PCA and MCA," *Int. J. Neural Syst.*, **7**, 53–67 (1996).
20. L. Xu, "Least mean square error reconstruction principle for self-organizing neural nets," *Neural Networks*, **6**, 627–648 (1993).
21. Q. Zhang and Y. Leung, "A class of learning algorithms for principal component analysis and minor component analysis," *IEEE Trans. on Neural Networks*, **11**, 529–533 (2000).
22. L. Ljung, "Analysis of recursive stochastic algorithms," *IEEE Trans. on Automatic Control*, **22**, 551–575 (1997).
23. T. Anisse and G. Cirrincione, "Against the convergence of the minor component analysis neurons," *IEEE Trans. on Neural Networks*, **10**, 207–210 (1999).
24. P. J. Zufiria, "On the discrete time dynamics of the basic hebbian neural network node," *IEEE Trans. on Neural Networks*, **13**, 1342–1352 (2002).
25. M. Ye, "Global convergence analysis of a self-stabilized MCA learning algorithm," *Neurocomputing*, **67**, 321–327 (2005).
26. M. Ye, "Robust beamforming by a globally convergent MCA neural network," *Lecture Notes in Computer Science*, **3496/2005**, 1034–1041 (2005).
27. M. Ye, X. -Q. Fan and X. Li, "A class of self-stabilizing MCA Learning algorithms," *IEEE Trans. on Neural Networks*, **17**, 1634–1638 (2006).
28. M. Ye, Z. Yi and K. K. Tan, "A robust and globally convergent PCA learning algorithm," *Control and Intelligent Systems*, **35**, 143–161 (2007).
29. Z. Yi, M. Ye, J. C. Lv and K. K. Tan, "Convergence analysis of a deterministic discrete time system of Oja's PCA learning algorithm," *IEEE Trans. on Neural Networks*, **16**, 1318–1328 (2005).
30. J. V. Stone, "Blind deconvolution using temporal predictability," *Neurocomputing*, **49**, 79–86 (2002).
31. C. Chatterjee, V. P. Roychowdhury, J. Ramos and M. D. Zoltowski, "Self-organizing algorithms for generalized eigen-decomposition," *IEEE Trans. on Neural Networks*, **8**, 1518–1530 (1997).
32. D. Peng and Z. Yi, "A modified Oja-Xu MCA learning algorithm and its convergence analysis," *IEEE Trans. on Circuits and Systems II, Exp. Briefs*, **54**, 348–352 (2007).
33. X. Y. Kong, C. H. Hu and C. Z. Han, "On the discrete time dynamics of a class of self-stabilizing MCA learning algorithms," *IEEE Trans. on Neural Networks*, **21**, 175–181 (2010).
34. D. Peng, Z. Yi and Y. Xiang, "A Unified Learning Algorithm to Extract Principal and Minor Components," *Digital Signal Processing*, **19**, 640–649 (2009).
35. D. Peng, Z. Yi and W. Luo, "Convergence Analysis of a Simple Minor Component Analysis Algorithm," *Neural Networks*, **20**, 842–850 (2007).
36. G. H. Golub and C. F. Van Loan, *Matrix Computation*, 3rd edn. (The Johns Hopkins Univ., 1996).
37. M. Ye, Y. Liu, H. Wu and Q. Liu, "A few on-line algorithms for extracting minor generalized eigenvectors," *Proc. IEEE International Conference on Neural Networks (Chengdu, China)*, 1715–1721 (2008).