Science in China Series F: Information Sciences

www.scichina.com
info.scichina.com

Adaptive blind separation of underdetermined mixtures based on sparse component analysis

YANG ZuYuan, HE ZhaoShui, XIE ShengLi† & FU YuLi

School of Electronics & Information Engineering, South China University of Technology, Guangzhou 510640, China

The independence *priori* is very often used in the conventional blind source separation (BSS). Naturally, independent component analysis (ICA) is also employed to perform BSS very often. However, ICA is difficult to use in some challenging cases, such as underdetermined BSS or blind separation of dependent sources. Recently, sparse component analysis (SCA) has attained much attention because it is theoretically available for underdetermined BSS and even for blind dependent source separation sometimes. However, SCA has not been developed very sufficiently. Up to now, there are only few existing algorithms and they are also not perfect as well in practice. For example, although Lewicki-Sejnowski's natural gradient for SCA is superior to K-mean clustering, it is just an approximation without rigorously theoretical basis. To overcome these problems, a new natural gradient formula is proposed in this paper. This formula is derived directly from the cost function of SCA through matrix theory. Mathematically, it is more rigorous. In addition, a new and robust adaptive BSS algorithm is developed based on the new natural gradient. Simulations illustrate that this natural gradient formula is more robust and reliable than Lewicki-Sejnowski's gradient.

underdetermined mixtures, blind source separation (BSS), dependent sources, sparse component analysis (SCA), sparse representation, independent component analysis (ICA), natural gradient

1 Introduction

Since blind source separation (BSS) has many potential applications in digital communication, speech and image processing, array signal processing, and medical signal processing, etc. It has become a hot spot in both signal processing and neural network fields. The mathematical model of BSS can be formulated as follows:

$$\boldsymbol{x} = \boldsymbol{A}\boldsymbol{s}, \tag{1}$$

Received November 11, 2006; accepted September 11, 2007
doi: 10.1007/s11432-008-0030-4
†Corresponding author (email: adshlxie@scut.edu.cn)
Supported by the National Natural Science Foundation of China (Grant Nos. 60505005, 60674033, 60774094 and U0635001), the Natural Science Fund of Guangdong Province, China (Grant Nos. 05103553 and 05006508), the Postdoctoral Science Foundation for Innovation from South China University of Technology, and China Postdoctoral Science Foundation (Grant No. 20070410237)

where $\boldsymbol{x}=(x_1,\cdots,x_m)^{\mathrm{T}}$ is the vector of observed signals, $\boldsymbol{s}=(s_1,\cdots,s_n)^{\mathrm{T}}$ is the vector of unknown sources, and $\boldsymbol{A}$ is the unknown $m\times n$ mixing matrix. The objective of BSS is to estimate source signals up to a scaling factor and a permutation factor only from the mixing signals.

The conventional BSS is usually based on the independence assumption[1,2]. For example, independent component analysis (ICA) based BSS[3−5] assumes that the sources are statistically mutually independent and requires that the number of sensors is not less than the number of sources, i.e., $m\geqslant n$. For nonsingular square matrix $\boldsymbol{A}$, the problem corresponds to estimating the inverse matrix $\boldsymbol{A}^{-1}$. Under this framework, many algorithms were proposed, e.g., information theory based algorithms (Bell-Sejnowski[6], Xie[7]), higher order statistics based algorithms (Hyvarinen[8−11], Comon[2]), non-stationarity based algorithms (Hyvarinen[12], Matsuoka[13]), and the geometrical algorithms[14]. However, for the underdetermined BSS or ill-posed BSS ($m<n$), the inverse matrix $\boldsymbol{A}^{-1}$ does not exist. The above methods are unavailable. Even if the mixing matrix $\boldsymbol{A}$ is given, one cannot obtain the unique solution in this case. Thus, new methods should be developed for estimating $\boldsymbol{A}$ and $\boldsymbol{s}$.

Furthermore, since the number of source signals is unknown and we usually have an only finite number of observed signals, sometimes the underdetermined mixtures occur in practice. Therefore, it is necessary to study the underdetermined models. Blind extraction was first proposed to perform underdetermined BSS, which extracts the source signals one by one. It should be noted that at most a part of sources, but not all, can be estimated by blind extraction for underdetermined BSS[15,16]. On the other hand, the precondition "statistical independence assumption" is not always reasonable in practice. For example, EEG, MEG, and FMRI sources are not statistically independent very often. BSS methods based on ICA are not available for these challenging problems because of the dependent sources.

For these reasons, the novel tools beyond ICA should be considered. Among them, the sparsity of source signals was recently exploited to achieve BSS by sparse representation or sparse decomposition. In this way, sparse component analysis (SCA) was directly or indirectly applied to underdetermined BSS, and promising results were reported[17]. For example, Belouchrani[18] proposed a maximum likelihood method, Zibulevsky[17] proposed a sparse decomposition method, overcomplete representation method was proposed by Lee[19], Lewicki[20], and Li[21], and the sparse representation in frequency domain was proposed by Bofill and Zebulevsky[8]. Typically, six source signals were successfully separated from only two mixing signals by Bofill and Zebulevsky[22]. In addition, SCA is even available for blind separation of dependent source signals sometimes (see Simulation 1 in this paper)[17].

As a whole, there are mainly two classes of BSS algorithms based on SCA. One of them is the so-called two-stage clustering-then-l^1-optimization algorithm, the other is adaptive algorithm based on overcomplete sparse representation. As its name implies, the former has two steps[17,21]: clustering analysis gives the estimation of the channel and linear programming estimates the source signals after the mixing matrix is estimated. Generally speaking, many clustering algorithms including K-mean are not very accurate and robust in practice. Thus, Lee, Lewicki et al.[17,23] developed the overcomplete representation based adaptive method, which can fill the gap of the former. Moreover, the results by Zhang, Amari, and Cichocki[24] further strengthen the adaptive method based on Lewicki-Sejnowski natural gradient[20]. However, Lewicki-Sejnowski natural gradient was derived by repeated approximations and is not rigorous mathematically.

Start from the cost function of SCA directly, we develop a rigorous natural gradient formula based on matrix theory. Also, an SCA based algorithm is presented for the underdetermined BSS in this paper. Simulations illustrate availability of this algorithm. Comparing with Lewicki-Sejnowski's natural gradient, the proposed natural gradient is more robust.

2 Sparse component analysis (SCA)

The SCA (or sparse representation) model is the same as that of BSS (1). However, different from ICA, SCA aims at estimating the basis matrix $\boldsymbol{A}$ and sparse components $\boldsymbol{s}$ such that $\boldsymbol{s}$ is as sparse as possible. In other words, $\boldsymbol{s}$ takes as many zeros as possible, or $\boldsymbol{s}$ has few nonzero entries if possible. Mathematically, this problem can be expressed as an l^0-norm optimization one[21,25],

$$(P_0)\ \min J(\boldsymbol{A},\boldsymbol{s}) = \min_{\boldsymbol{A},\boldsymbol{s}} \|\boldsymbol{s}\|^0 = \min_{\boldsymbol{A},\boldsymbol{s}} \sum_{i=1}^{n} |s_i|^0,\ \text{ st: } \boldsymbol{A}\boldsymbol{s} = \boldsymbol{x}, \tag{2}$$

where $\|\boldsymbol{s}\|^0$ measures the number of nonzero entries in the vector $\boldsymbol{s}$[21,25]. Here $|s_i|^0 = 1$ for $s_i \neq 0$, while $|s_i|^0 = 0$ if and only if $s_i = 0$.

Although the problem (2) can give the sparsest solution, it has no unique solution in general. Furthermore, it is very difficult to solve problem (2), and its solution is not robust but is very sensitive to noise. For this reason, Donoho[25] and Li[21] suggested using l^1-norm to measure the sparsity. Thus, the l^0-norm optimization problem (2) is replaced by its approximation——an l^1-norm optimization problem

$$(P_1)\ \min J(\boldsymbol{A},\boldsymbol{s}) = \min_{\boldsymbol{A},\boldsymbol{s}} \|\boldsymbol{s}\|^1 = \min_{\boldsymbol{A},\boldsymbol{s}} \sum_{i=1}^{n} |s_i|,\ \text{ st: } \boldsymbol{A}\boldsymbol{s} = \boldsymbol{x}. \tag{3}$$

Donoho and Li found that the solutions to (3) are good approximations of the solutions to (2), when the source $\boldsymbol{s}$ is sufficiently sparse. Furthermore, there are many existing mathematical tools for l^1-norm optimization problem. So we are interested to achieve SCA by solving problem (3). An example was given by Zibulevsky and Pearlmutter[17].

In the engineering applications, many signals (e.g., speech signals or other signals detected by some instruments) are with limited frequency bandwidths and are a little smooth in some degree (but maybe are not very smooth), which results in that these signals (except white Gaussian noise) in the real world are sparse in some degree in the time domain, or they can be sparsely represented in an appropriate transform domain (e.g., frequency domain or wavelet domain). Thus, SCA has many applications in practice[15,17]. Especially, it can be applied to BSS. If the source signals are sufficiently sparse, SCA can be applied to underdetermined BSS or BSS of dependent sources (see simulation 1 in this paper)[15,17,19,21,22]. SCA can achieve better result than the conventional ICA even for the well-deterdetermined BSS[17,19]. To well satisfy the sparseness assumption, if the source signals are not sparse in time domain, we can consider to perform BSS in an appropriate transform domain by Fourier transform, wavelet or other transforms.

Next, we discuss how to perform SCA by minimizing l^1-norm, and here we mainly discuss how solve problem (3).

3 Natural gradient of mixing matrix *A*

To identify mixing matrix $\boldsymbol{A}$, Lewicki and Sejnowski[20] originally presented an approximate natural gradient with respect to the mixing matrix $\boldsymbol{A}$ based on "overcomplete representation

learning" theory. Later, Lee and Lewicki[19] successfully identified $\boldsymbol{A}$ using this approximate gradient. Here start from the optimization problem (3), we will derive a new and rigorous natural gradient for learning the mixing matrix $\boldsymbol{A}$. For this problem, we have the following theorem:

Theorem 1. For the optimization problem (3), the natural gradient of cost function $J(\cdot)$ with respect to $\boldsymbol{A}$ can be expressed as follows:

$$\nabla J(\boldsymbol{A}) = -\boldsymbol{A} \cdot (\operatorname{sign}(\boldsymbol{s}) \cdot \boldsymbol{s}^{\mathrm{T}}), \tag{4}$$

where $\operatorname{sign}(\boldsymbol{s}) = (\operatorname{sign}(s_1), \cdots, \operatorname{sign}(s_n))^{\mathrm{T}}$, and $\operatorname{sign}(\cdot)$ is a sign function.

Proof. In the optimization problem (3), the cost function $J = \sum_{i=1}^{n} |s_i|$ is not differentiable at point 0. Thus, we introduce an auxiliary function $\rho(x, \delta)$

$$\rho(x, \delta) = \begin{cases} |x|, & |x| \geqslant \delta, \\ \dfrac{x^2}{2\delta} + \dfrac{\delta}{2}, & |x| \leqslant \delta, \end{cases} \tag{5}$$

where δ is a small positive number. The function $\rho(x, 0.1)$ is shown in Figure 1.

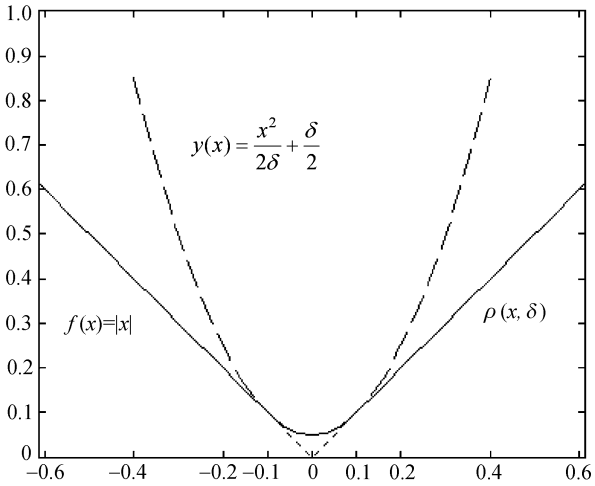

Figure 1 Function $\rho(x, \delta)$ when $\delta = 0.1$.

Function $\rho(x, \delta)$ has the following properties:

1) $\rho(x, \delta) \geqslant |x|$, for $\delta \to 0$; $\rho(x, \delta)$ uniformly converges to $|x|$, i.e., $\lim_{\delta \to 0} \rho(x, \delta) = |x|$, $x \in (-\infty, +\infty)$;

2) $\rho(x, \delta)$ is differentiable with respect to x, and its derivative is as follows:

$$\phi(x, \delta) = \frac{d\rho}{dx} = \begin{cases} \operatorname{sign}(x), & |x| \geqslant \delta, \\ \dfrac{x}{\delta}, & |x| \leqslant \delta, \end{cases} \tag{6}$$

where $\phi(0, \delta) = 0$ and $\phi(x, \delta)$ converges to $\operatorname{sign}(x)$ almost everywhere, i.e., $\forall x \in (-\infty, +\infty)$, $\lim_{\delta \to 0} \phi(x, \delta) = \operatorname{sign}(x)$.

By the function $\rho(x, \delta)$, problem (3) can be approximated by the following optimization one:

$$\min L(\boldsymbol{A},\boldsymbol{s}) = \min_{\boldsymbol{A},\boldsymbol{s}} \sum_{i=1}^{n} \rho(s_i,\delta), \quad \text{st:} \quad \boldsymbol{A}\boldsymbol{s} = \boldsymbol{x}, \tag{7}$$

when $\delta \to 0$, problem (7) tends to problem (3). Since function $\rho(x,\delta)$ is differentiable with respect to x, we can compute the derivative of $L(\cdot)$ with respect to x.

We prove Theorem 1 for two cases: $m=n$ ($\boldsymbol{A}$ is nonsingular), and $m < n$ ($\boldsymbol{A}$ is overcomplete).

Case 1: $m=n$ and $\boldsymbol{A}$ is nonsingular

Since $\boldsymbol{A}$ is nonsingular, suppose that $\boldsymbol{A} = \boldsymbol{W}^{-1}$. From (1), we have

$$\boldsymbol{s} = \boldsymbol{W}\boldsymbol{x}. \tag{8}$$

Then, the optimization problem (7) can be changed into

$$\min L(\boldsymbol{W},\delta) = \min_{\boldsymbol{W}} \sum_{i=1}^{n} \rho(s_i,\delta), \quad \text{st:} \quad \boldsymbol{s} = \boldsymbol{W}\boldsymbol{x}. \tag{9}$$

The derivative of $L(\cdot,\delta)$ with respect to w_{ij} is

$$\frac{\partial L(\cdot,\delta)}{\partial w_{ij}} = \sum_{k=1}^{n} \frac{\partial L(\cdot,\delta)}{\partial s_k} \cdot \frac{\partial s_k}{\partial w_{ij}} = \frac{\partial L(\cdot,\delta)}{\partial s_i} \cdot \frac{\partial s_i}{\partial w_{ij}} = \phi(s_i,\delta) \cdot x_j, \ i,j = 1,\cdots,n.$$

That is,

$$\frac{\partial L(\cdot,\delta)}{\partial \boldsymbol{W}} = \phi(\boldsymbol{s},\delta) \cdot \boldsymbol{x}^{\mathrm{T}}, \tag{10}$$

where $\phi(\boldsymbol{s},\delta) = (\phi(s_1,\delta),\cdots,\phi(s_n,\delta))^{\mathrm{T}}$. The function $\phi(\cdot,\delta)$ is defined as (6). Since $\boldsymbol{A} = \boldsymbol{W}^{-1}$, the gradient of $L(\cdot,\delta)$ with respect to $\boldsymbol{A}$ can be written as (see Appendix),

$$\frac{\partial L(\cdot,\delta)}{\partial \boldsymbol{A}} = -\boldsymbol{A}^{-\mathrm{T}} \cdot \frac{\partial L(\cdot,\delta)}{\partial \boldsymbol{W}} \cdot \boldsymbol{A}^{-\mathrm{T}} = -\boldsymbol{A}^{-\mathrm{T}} \cdot \phi(\boldsymbol{s},\delta) \cdot \boldsymbol{x}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}} = -\boldsymbol{A}^{-\mathrm{T}} \cdot \phi(\boldsymbol{s},\delta) \cdot \boldsymbol{s}^{\mathrm{T}}.$$

Thus, the natural gradient of $L(\cdot,\delta)$ with respect to $\boldsymbol{A}$ is

$$\nabla L(\boldsymbol{A},\delta) = \boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} \cdot \frac{\partial L(\cdot,\delta)}{\partial \boldsymbol{A}} = -\boldsymbol{A}\boldsymbol{A}^{\mathrm{T}} \cdot \boldsymbol{A}^{-\mathrm{T}} \cdot \phi(\boldsymbol{s},\delta) \cdot \boldsymbol{s}^{\mathrm{T}} = -\boldsymbol{A} \cdot \phi(\boldsymbol{s},\delta) \cdot \boldsymbol{s}^{\mathrm{T}}. \tag{11}$$

Note that there is no inverse matrix $\boldsymbol{A}^{-1}$ in the natural gradient $\nabla L(\boldsymbol{A},\delta)$ in (11). So the natural gradient $\nabla L(\boldsymbol{A},\delta)$ is simpler than standard gradient $\partial L(\cdot,\delta)/\partial \boldsymbol{A}$. From the Lemma 2 in Appendix, this natural gradient can be used to solve optimization problems (7). When $\delta \to 0$, problem (7) approximates to problem (3), $L(\cdot,\delta) \to J(\cdot)$ and $\nabla L(\boldsymbol{A},\delta)$ tends to $\nabla J(\boldsymbol{A})$ almost everywhere, i.e.,

$$\forall \boldsymbol{s}, \ \lim_{\delta \to 0} \nabla L(\boldsymbol{A},\delta) = \nabla J(\boldsymbol{A}) = -\boldsymbol{A} \cdot (\mathrm{sign}(\boldsymbol{s}) \cdot \boldsymbol{s}^{\mathrm{T}}). \tag{12}$$

Thus, the natural gradient of $J(\cdot)$ with respect to $\boldsymbol{A}$ is $\nabla J(\boldsymbol{A}) = -\boldsymbol{A} \cdot (\mathrm{sign}(\boldsymbol{s}) \cdot \boldsymbol{s}^{\mathrm{T}})$. Theorem 1 is proven in Case 1.

Case 2: $m< n$

In this case, $\boldsymbol{A}$ is an $m\times n$ matrix. We add $n-m$ virtual observed signals $\boldsymbol{x}^{\mathrm{virtual}}_{(n-m)\times 1}$. The model (1) can be extended as

$$\boldsymbol{x}^{\mathrm{new}} = \boldsymbol{A}^{\mathrm{new}}\boldsymbol{s}. \tag{13}$$

It is possible to add the virtual observed signals such that $\boldsymbol{A}^{\mathrm{new}}$ is an $n\times n$ nonsingular matrix,

where $\boldsymbol{x}^{\text{new}} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{x}^{\text{virtual}}_{(n-m)\times 1} \end{pmatrix}$, and $\boldsymbol{A}^{\text{new}} = \begin{pmatrix} \boldsymbol{A} \\ \overline{\boldsymbol{A}}^{\text{virtual}}_{(n-m)\times n} \end{pmatrix}$. The sub-matrix $\overline{\boldsymbol{A}}^{\text{virtual}}_{(n-m)\times n}$ associates with $\boldsymbol{x}^{\text{virtual}}_{(n-m)\times 1}$. Since $\boldsymbol{A}^{\text{new}}$ is nonsingular, from the proof of Case 1, the natural gradient of $J(\cdot)$ with respect to $\boldsymbol{A}^{\text{new}}$ is analogously as follows:

$$\nabla J(\boldsymbol{A}^{\text{new}}) = \begin{pmatrix} \nabla J(\boldsymbol{A}) \\ \nabla J(\overline{\boldsymbol{A}}^{\text{virtual}}_{(n-m)\times n}) \end{pmatrix} = -\boldsymbol{A}^{\text{new}} \cdot \text{sign}(\boldsymbol{s}) \cdot \boldsymbol{s}^{\text{T}}$$

$$= -\begin{pmatrix} \boldsymbol{A} \\ \overline{\boldsymbol{A}}^{\text{virtual}}_{(n-m)\times n} \end{pmatrix} \cdot \text{sign}(\boldsymbol{s}) \cdot \boldsymbol{s}^{\text{T}} = \begin{pmatrix} -\boldsymbol{A} \cdot \text{sign}(\boldsymbol{s}) \cdot \boldsymbol{s}^{\text{T}} \\ -\overline{\boldsymbol{A}}^{\text{virtual}}_{(n-m)\times n} \cdot \text{sign}(\boldsymbol{s}) \cdot \boldsymbol{s}^{\text{T}} \end{pmatrix},$$

or, it can be re-written as

$$\nabla J(\boldsymbol{A}^{\text{new}}) = \begin{pmatrix} \nabla J(\boldsymbol{A}) \\ \nabla J(\overline{\boldsymbol{A}}^{\text{virtual}}_{(n-m)\times n}) \end{pmatrix} = \begin{pmatrix} -\boldsymbol{A} \cdot \text{sign}(\boldsymbol{s}) \cdot \boldsymbol{s}^{\text{T}} \\ -\overline{\boldsymbol{A}}^{\text{virtual}}_{(n-m)\times n} \cdot \text{sign}(\boldsymbol{s}) \cdot \boldsymbol{s}^{\text{T}} \end{pmatrix}.$$

Thus, for $m < n$, the natural gradient of $J(\cdot)$ with respect to $\boldsymbol{A}$ also can be formulated as

$$\nabla J(\boldsymbol{A}) = -\boldsymbol{A} \cdot (\text{sign}(\boldsymbol{s}) \cdot \boldsymbol{s}^{\text{T}}).$$

The natural gradient (4) is very simple and effective, which can be employed to perform SCA. In the next section, we will establish a BSS algorithm based on SCA.

4 BSS algorithm

By the above natural gradient, we have the following iterative rule for learning the mixing matrix $\boldsymbol{A}$:

$$\boldsymbol{A}^{(k+1)} = \boldsymbol{A}^{(k)} - \mu \nabla J(\boldsymbol{A}^{(k)}) = \boldsymbol{A}^{(k)} + \mu \boldsymbol{A}^{(k)} \cdot \text{sign}(\boldsymbol{s}^{(k)}) \cdot (\boldsymbol{s}^{(k)})^{\text{T}}, \tag{14}$$

where $\mu > 0$. As pointed out in ref. [17], the solution to problem (3) will be $\boldsymbol{s} \approx 0$, if there are no additional constraints. Obviously, this is not the right solution. For this point, we add some additional constraints to problem (3)[21,26]: We constrain the l^2-norm of each column of $\boldsymbol{A}$ to be 1[26], i.e., $\|\boldsymbol{a}_i\| = 1,\ i = 1, \cdots, n$. To this end, we consider a similar method to Parra and Spence's[27] constrained gradient. In formula (14), instead of natural gradient $\nabla J(\boldsymbol{A})$, we consider the projection of $\nabla J(\boldsymbol{A})$ on the hyperplanes defined by $\|\boldsymbol{a}_i\| = 1,\ i = 1, \cdots, n$. The projection operator for the ith column $\nabla J(\boldsymbol{a}_i)$ of $\nabla J(\boldsymbol{A}) = [\nabla J(\boldsymbol{a}_1), \cdots, \nabla J(\boldsymbol{a}_n)]$ is

$$P_i = \boldsymbol{I} - \boldsymbol{a}_i \cdot \boldsymbol{a}_i^{\text{T}}, \text{ or } P: \nabla J(\boldsymbol{a}_i) \to (\boldsymbol{I} - \boldsymbol{a}_i \cdot \boldsymbol{a}_i^{\text{T}}) \cdot \nabla J(\boldsymbol{a}_i). \tag{15}$$

We can also re-write it as a constrained natural gradient

$$\nabla J(\boldsymbol{A})\Big|_{\|\boldsymbol{a}_i\|=1,\ i=1,\cdots,n} = \nabla J(\boldsymbol{A}) - \boldsymbol{A} \cdot \text{diag}(\boldsymbol{A}^{\text{T}} \cdot \nabla J(\boldsymbol{A})). \tag{16}$$

The constrained natural gradient (16) $\nabla J(\boldsymbol{A})\Big|_{\|\boldsymbol{a}_i\|=1,\ i=1,\cdots,n}$ is not always necessary. From numerical experiments, we find that good performance also can be achieved generally if we combine natural gradient $\nabla J(\boldsymbol{A})$ with the column normalization operators.

In formula (14), we cannot update $\boldsymbol{A}^{(k)}$ unless $\boldsymbol{s}^{(k)}$ is given. To achieve this goal, we can

alternatively update $\boldsymbol{A}$ and $\boldsymbol{s}$ in a similar way in ref. [19]. If the mixing matrix $\boldsymbol{A}$ is known, the optimization problem (3) can be simplified as

$$\begin{cases} \min J(\boldsymbol{s} \mid \boldsymbol{A}) = \min\limits_{\boldsymbol{s}} \sum_{i=1}^{n} |s_i|, \\ \text{st}: \boldsymbol{A}\boldsymbol{s} = \boldsymbol{x}. \end{cases} \tag{17}$$

The shortest path decomposition (SPD) can be employed to solve the optimization problem (17)[22,26]. For simplicity, we denotes the solution of (17) as $\hat{\boldsymbol{s}}^* = \text{SPD}(\boldsymbol{A}, \boldsymbol{x})$.

From the above discussion, the BSS algorithm for underdetermined mixtures based on SCA can be outlined as follows:

1) Initialize $\boldsymbol{A}$ as $\boldsymbol{A}^{(0)}$, where each column of $\boldsymbol{A}^{(0)}$ is normalized such that its l^2-norm is "1". Set the step size $\mu > 0$ and let $k = 0$.

2) Estimate $\hat{\boldsymbol{s}}^{(k)} = \text{SPD}(\boldsymbol{A}^{(k)}, \boldsymbol{x})$ by SPD.

3) Substituting $\hat{\boldsymbol{s}}^{(k)}$ into (14), compute the constrained natural gradient (16) $\nabla J(\boldsymbol{A})\|_{\|\boldsymbol{a}_i\|=1,\ i=1,\cdots,n}$. Update $\boldsymbol{A}$ as $\boldsymbol{A}^{(k+1)}$. Let $k = k+1$.

4) If the algorithm is convergent, turn to 5), otherwise, go to 2).

5) Output the estimation of the source signals $\boldsymbol{s}^* = \hat{\boldsymbol{s}}^{(k)}$.

5 Simulations

To evaluate performance, two performance indices are used here. One is the bias angle $ae(\boldsymbol{a}, \hat{\boldsymbol{a}})$ between the column $\boldsymbol{a}$ in mixing matrix and its estimation $\hat{\boldsymbol{a}}$. It is defined as

$$ae(\boldsymbol{a}, \hat{\boldsymbol{a}}) = \frac{180}{\pi} \text{arcCos}\left(\frac{\langle \boldsymbol{a}, \hat{\boldsymbol{a}} \rangle}{\|\boldsymbol{a}\| \cdot \|\hat{\boldsymbol{a}}\|} \right), \tag{18}$$

$ae(\boldsymbol{a}, \hat{\boldsymbol{a}})$ is small if the direction of $\boldsymbol{a}$ is close to that of $\hat{\boldsymbol{a}}$. Otherwise, $ae(\boldsymbol{a}, \hat{\boldsymbol{a}})$ is large.

Another index is signal to interference ratio (SIR) between separated signal $\hat{s}$ and source signal s[22].

$$\text{SIR}(s, \hat{s}) = 10 \log[s^2 / (s - \hat{s})^2] \ \text{(dB)}. \tag{19}$$

Since separated signal $\hat{s}$ can be different from s up to a scaling factor c $(c \neq 0)$, as treated in ref. [22], we adjust s and $\hat{s}$ to be with the equal energy, before the SIR is computed.

Simulation 1. SCA based BSS for dependent signals

Consider two source signals $s_1(t)$ and $s_2(t), t = 1, \cdots, 1000$. All the samples of $s_1(t)$ and $s_2(t)$ take values from the binary set {0, 1}, i.e., $s_1(t) = 1$ or $s_1(t) = 0$, and $s_2(t) = 1$ or $s_2(t)=0$. In addition, we constrain that $s_1(t) + s_2(t) = 1,\ t = 1, \cdots, 1000$. Obviously, $s_1(t)$ and $s_2(t)$ are statistically dependent because they satisfy the equation $s_1(t) = 1 - s_2(t),\ t = 1, \cdots, 1000$. The mixing matrix is generated by random as

$$\boldsymbol{A} = \begin{pmatrix} 0.9673 & -0.9105 \\ 0.2537 & 0.4135 \end{pmatrix}.$$

From model (1), we have observed signals $x_1(t)$ and $x_2(t)$.

Set the step $\mu = 0.01$. The algorithm converged after 500 iterations. The mixing matrix $\boldsymbol{A}$ is estimated as follows:

$$\hat{\boldsymbol{A}} = \begin{pmatrix} 0.9671 & -0.9093 \\ 0.2543 & 0.4160 \end{pmatrix}.$$

By calculation, we have $ae(\boldsymbol{a}_1, \hat{\boldsymbol{a}}_1) = 0.036$ and $ae(\boldsymbol{a}_2, \hat{\boldsymbol{a}}_2) = 0.16$. The SIRs of the signals are 151.7109 dB and 109.8398 dB. Thus, two dependent signals were successfully separated.

Simulation 2. The performance comparison between the constrained natural gradient (16) and the natural gradient presented by Lewicki-Sejnowski.

Suppose that the signals follow Laplacian distribution. From ref. [19], Lewicki-Sejnowski's natural gradient is $\Delta \boldsymbol{A} \propto -\boldsymbol{A} \cdot (-sign(\boldsymbol{s}) \cdot \boldsymbol{s}^{\mathrm{T}} + \boldsymbol{I})$, in which the identity matrix $\boldsymbol{I}$ prevents the separated signals tending to zero. The projection gradient presented in this paper plays the same role.

For the BSS example of "two observed signals and three source signals" in ref. [13], simulations show that both our natural gradient and Lewicki-Sejnowski's natural gradient exhibited good performance. To further compare their performance, Bofill and Zibulevsky's[22] example of "two observed signals mixed by six source signals" is taken here.

The sources are six flute signals. There are a total of 32768 samples in each signal. Figure 2(a) shows the waveforms of them.

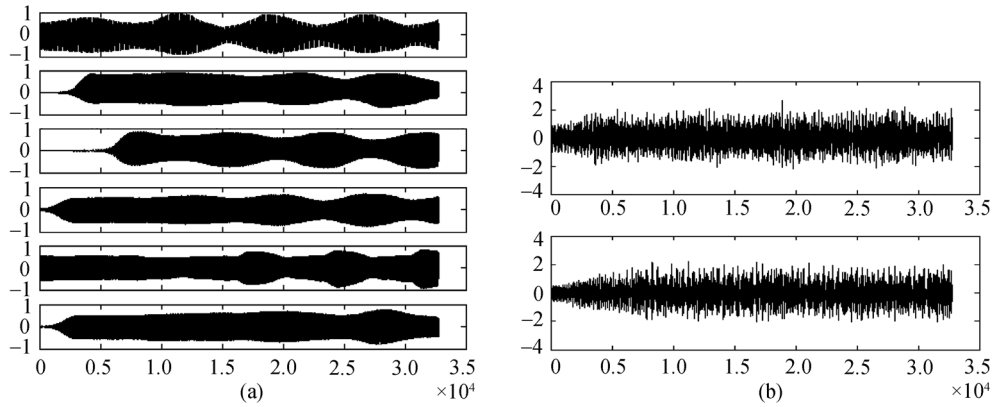

Figure 2 (a) The six source signals; (b) the two mixed signals.

Taking the same mixing matrix as in ref. [8],

$$\boldsymbol{A} = \begin{pmatrix} 0.9659 & 0.7071 & 0.2588 & -0.2588 & -0.7071 & -0.9659 \\ 0.2588 & 0.7071 & 0.9659 & 0.9659 & 0.7071 & 0.2588 \end{pmatrix},$$

the mixing signals are shown in Figure 2(b). For the two natural gradients, the same initialization was taken to update $\boldsymbol{A}$. Extensive tests show that Lewicki-Sejnowski's natural gradient failed to achieve BSS. For example, if the initialization is taken randomly as follows:

$$\boldsymbol{A}^{(0)} = \begin{pmatrix} -0.2514 & 0.3994 & -0.6935 & 0.9995 & 0.8823 & -0.2491 \\ -0.9679 & 0.9168 & 0.7204 & -0.0316 & 0.4708 & 0.9685 \end{pmatrix},$$

and the step size is set as $\mu = 0.01$, the algorithm that uses gradient (16) is convergent after 100

iterations. The algorithm by Lewicki-Sejnowski's natural gradient also converges after more than 200 iterations. Let the estimations of $\boldsymbol{A}$ be $\hat{\boldsymbol{A}}_{\text{Our}}$ and $\hat{\boldsymbol{A}}_{\text{LS}}$, respectively. We have

$$\hat{\boldsymbol{A}}_{\text{Our}} = \begin{pmatrix} 0.9667 & 0.7096 & 0.2579 & -0.2597 & -0.7055 & -0.9657 \\ 0.2560 & 0.7046 & 0.9662 & 0.9657 & 0.7087 & 0.2595 \end{pmatrix},$$

$$\hat{\boldsymbol{A}}_{\text{LS}} = \begin{pmatrix} 6.1008 & 3.2397 & 0.0000 & -1.3476 & -4.5548 & -0.0620 \\ 1.4805 & 4.2478 & 0.0000 & 5.4940 & 3.8266 & 0.0165 \end{pmatrix}.$$

In this case, the bias angles and SIRs are given in Tables 1 and 2.

Table 1 The bias angles of each pair of columns of $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}$

Simulation 2	The angles (°)					
	$ae(\boldsymbol{a}_1, \hat{\boldsymbol{a}}_1)$	$ae(\boldsymbol{a}_2, \hat{\boldsymbol{a}}_2)$	$ae(\boldsymbol{a}_3, \hat{\boldsymbol{a}}_3)$	$ae(\boldsymbol{a}_4, \hat{\boldsymbol{a}}_4)$	$ae(\boldsymbol{a}_5, \hat{\boldsymbol{a}}_5)$	$ae(\boldsymbol{a}_6, \hat{\boldsymbol{a}}_6)$
Lewicki-Sejnowski's natural gradient	1.3597	7.6681	10.1678	1.2179	4.9657	0.1005
Constrained natural gradient (16)	0.1663	0.2024	0.0570	0.0521	0.1311	0.0399

Table 2 The SIRs of the separated signals

Simulation 2	SIR (dB)					
Source signals	1	2	3	4	5	6
Lewicki-Sejnowski's natural gradient	12.8135	9.0512	0	14.3304	10.4519	0
Constrained natural gradient (16)	50.5133	52.1840	49.1864	43.4974	49.1196	51.9210

From Tables 1, 2, and Figure 3, we can see that Lewicki-Sejnowski's natural gradient failed to achieve BSS, while the constrained natural gradient (16) succeeded.

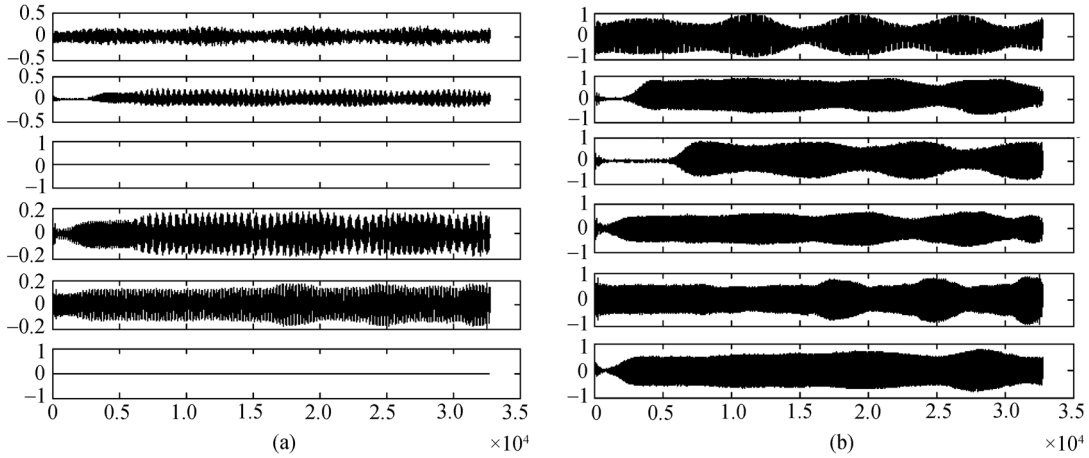

Figure 3 The separated signals by two natural gradients. (a) The separated signals by Lewicki-Sejnowski's gradient; (b) the separated signals by the constrained natural gradient (16).

Simulation 3. The robustness test of algorithm in noisy environment

The same source signals and mixing matrix are taken as in Simulation 2. However, 10 dbw white Gaussian noise was added to the observed signals here. We check the robustness of our algorithm in the noisy environment. The observed signals corrupted by noise are shown in Figure 4.

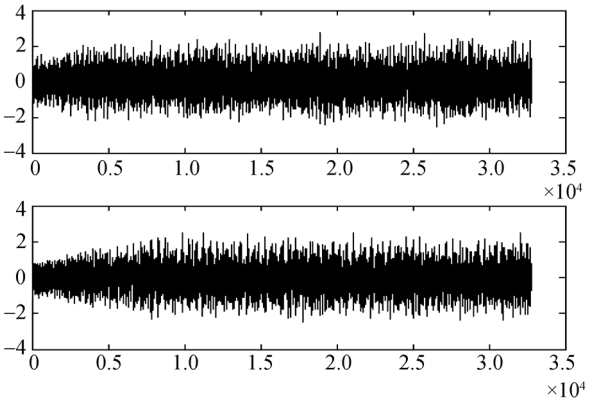

Figure 4 The observed signals with noise.

The learning step is set as $\mu = 0.01$. After 50 iterations, the algorithm converged and obtained the estimation of mixing matrix $\boldsymbol{A}$ as follows:

$$\hat{\boldsymbol{A}} = \begin{pmatrix} 0.2557 & 0.7069 & -0.7093 & -0.9665 & 0.9670 & -0.2612 \\ 0.9668 & 0.7073 & 0.7049 & 0.2567 & 0.2549 & 0.9653 \end{pmatrix}.$$

The bias angles and SIRs are shown in Tables 3 and 4, respectively. Figure 5 shows the separated signals in noisy case.

Table 3 The bias angles of each pair of columns of $\boldsymbol{A}$ and $\hat{\boldsymbol{A}}$ in noisy case

Bias angles $ae(\boldsymbol{a},\hat{\boldsymbol{a}})$ (°)					
$ae(\boldsymbol{a}_1,\hat{\boldsymbol{a}}_1)$	$ae(\boldsymbol{a}_2,\hat{\boldsymbol{a}}_2)$	$ae(\boldsymbol{a}_3,\hat{\boldsymbol{a}}_3)$	$ae(\boldsymbol{a}_4,\hat{\boldsymbol{a}}_4)$	$ae(\boldsymbol{a}_5,\hat{\boldsymbol{a}}_5)$	$ae(\boldsymbol{a}_6,\hat{\boldsymbol{a}}_6)$
0.2333	0.0144	0.1842	0.1393	0.1771	0.1282

Table 4 The SIRs of the estimations by the proposed natural gradient in noisy case

Simulation 3	SIRs (dB)					
Source signals	1	2	3	4	5	6
Estimated signals	24.8822	25.0302	25.1542	24.2955	24.8437	25.0588

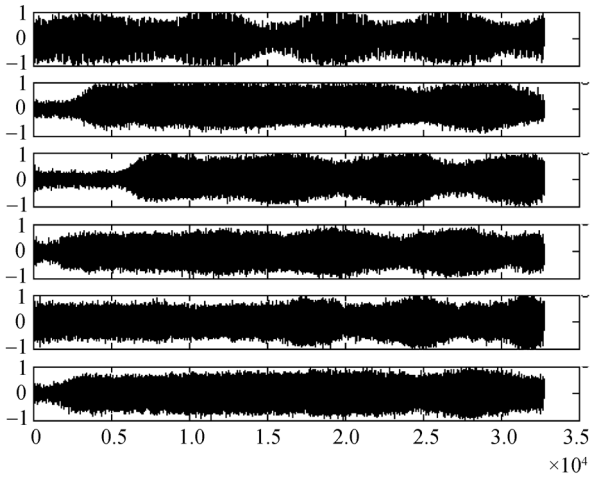

Figure 5 The separated signals (in noisy case).

Table 3 shows that although the heavy noise was added to the mixtures (10 dbw), the mixing matrix was very precisely estimated. Table 4 shows the SIRs of the separated signals. All SIRs are greater than 24 dB.

6 Conclusion

Most of BSS methods are based on the conventional ICA, which are unavailable for some challenging problems such as underdetermined BSS or blind separation of dependent sources. To face these challenges, SCA method is introduced in this paper. Also, this paper presents a new natural gradient and rigorously proves it by matrix theory. In addition, this natural gradient is successfully applied to SCA. The developed SCA can be used to perform underdetermined BSS if the sources are sufficiently sparse. Furthermore, our method is a potential tool for blind separation of dependent sources.

Appendix

Lemma 1. Let $\boldsymbol{A}$ be an $n\times n$ invertible matrix and $\boldsymbol{W}$ be the inverse of $\boldsymbol{A}$, i.e., $\boldsymbol{W}=\boldsymbol{A}^{-1}$. If $J(\boldsymbol{A})$ is a differentiable with respect to $\boldsymbol{A}$ (with $n\times n$ variables), then the partial derivatives of J with respect to matrix $\boldsymbol{A}$ and the inverse $\boldsymbol{W}$ satisfy the following relation:

$$\frac{\partial J}{\partial \boldsymbol{A}}=-\boldsymbol{A}^{-\mathrm{T}}\frac{\partial J}{\partial \boldsymbol{W}}\boldsymbol{A}^{-\mathrm{T}}.$$

Proof. Obviously, we have

$$\boldsymbol{A}^{\mathrm{T}}\boldsymbol{W}^{\mathrm{T}}=\boldsymbol{I}_{n\times n}. \tag{A1}$$

Let a_{ij} be the ith row and jth column of $\boldsymbol{A}$. We have

$$\begin{aligned}&\frac{\partial \boldsymbol{A}^{\mathrm{T}}}{\partial a_{ij}}\boldsymbol{W}^{\mathrm{T}}+\boldsymbol{A}^{\mathrm{T}}\frac{\partial \boldsymbol{W}^{\mathrm{T}}}{\partial a_{ij}}=\boldsymbol{0}_{n\times n}\Rightarrow\\ &\frac{\partial \boldsymbol{W}^{\mathrm{T}}}{\partial a_{ij}}=-\boldsymbol{A}^{-\mathrm{T}}\frac{\partial \boldsymbol{A}^{\mathrm{T}}}{\partial a_{ij}}\boldsymbol{W}^{\mathrm{T}}=-\boldsymbol{A}^{-\mathrm{T}}\frac{\partial \boldsymbol{A}^{\mathrm{T}}}{\partial a_{ij}}\boldsymbol{A}^{-\mathrm{T}}.\end{aligned} \tag{A2}$$

On the other hand, we have

$$\begin{aligned}\frac{\partial J}{\partial a_{ij}}&=Trace\left\{\frac{\partial J}{\partial \boldsymbol{W}}\frac{\partial \boldsymbol{W}^{\mathrm{T}}}{\partial a_{ij}}\right\}\\ &=-Trace\left\{\left(\frac{\partial J}{\partial \boldsymbol{W}}\boldsymbol{A}^{-\mathrm{T}}\frac{\partial \boldsymbol{A}^{\mathrm{T}}}{\partial a_{ij}}\right)\boldsymbol{A}^{-\mathrm{T}}\right\}=-Trace\left\{\left(\boldsymbol{A}^{-\mathrm{T}}\frac{\partial J}{\partial \boldsymbol{W}}\boldsymbol{A}^{-\mathrm{T}}\right)\frac{\partial \boldsymbol{A}^{\mathrm{T}}}{\partial a_{ij}}\right\},\end{aligned} \tag{A3}$$

where $\dfrac{\partial \boldsymbol{A}^{\mathrm{T}}}{\partial a_{ij}}$ is an $n\times n$ matrix in which all entries are zero except the one at jth row and ith column. Thus, we have

$$Trace\left\{\left(\boldsymbol{A}^{-\mathrm{T}}\frac{\partial J}{\partial \boldsymbol{W}}\boldsymbol{A}^{-\mathrm{T}}\right)\frac{\partial \boldsymbol{A}^{\mathrm{T}}}{\partial a_{ij}}\right\}=\left(\boldsymbol{A}^{-\mathrm{T}}\frac{\partial J}{\partial \boldsymbol{W}}\boldsymbol{A}^{-\mathrm{T}}\right)_{ij}. \tag{A4}$$

From (A3) and (A4), we can derive

$$\frac{\partial J}{\partial a_{ij}} = \left(\frac{\partial J}{\partial \boldsymbol{A}}\right)_{ij} = -\left(\boldsymbol{A}^{-\mathrm{T}} \frac{\partial J}{\partial \boldsymbol{W}} \boldsymbol{A}^{-\mathrm{T}}\right)_{ij}.$$

Hence, we obtain the conclusion

$$\frac{\partial J}{\partial \boldsymbol{A}} = -\boldsymbol{A}^{-\mathrm{T}} \frac{\partial J}{\partial \boldsymbol{W}} \boldsymbol{A}^{-\mathrm{T}}.$$

Lemma 2. Let $L(\cdot)$ be a differentiable function with respect to $\boldsymbol{A} \in R^{m\times n}$. Consider the following iterative formula

$$\boldsymbol{A}^{(k+1)} = \boldsymbol{A}^{(k)} + \mu \nabla L(\boldsymbol{A}^{(k)}), \tag{A5}$$

where $\nabla L(\boldsymbol{A}) = \boldsymbol{U}\boldsymbol{U}^T \frac{\partial L}{\partial \boldsymbol{A}}$, and $\boldsymbol{U}$ is an arbitrary matrix with m rows. If the learning step $\mu > 0$ is sufficiently small, the following inequality holds

$$L(\boldsymbol{A}^{(k+1)}) \geqslant L(\boldsymbol{A}^{(k)}).$$

Proof. Taking the first Taylor extension of $L(\boldsymbol{A})$, one has

$$L(\boldsymbol{A}+\boldsymbol{\varepsilon}) = L(\boldsymbol{A}) + \left\langle \frac{\partial L}{\partial \boldsymbol{A}}, \boldsymbol{\varepsilon} \right\rangle + o(\boldsymbol{\varepsilon}) = L(\boldsymbol{A}) + Trace\left(\left(\frac{\partial L}{\partial \boldsymbol{A}}\right)^{\mathrm{T}} \cdot \boldsymbol{\varepsilon}\right) + o(\boldsymbol{\varepsilon}). \tag{A6}$$

Substituting $\boldsymbol{\varepsilon} = \mu \nabla L(\boldsymbol{A}) = \mu \boldsymbol{U}\boldsymbol{U}^T \frac{\partial L}{\partial \boldsymbol{A}}$ into (A6), we obtain

$$\begin{aligned} L(\boldsymbol{A}+\boldsymbol{\varepsilon}) &= L(\boldsymbol{A}) + Trace\left(\left(\frac{\partial L}{\partial \boldsymbol{A}}\right)^{\mathrm{T}} \cdot \mu \boldsymbol{U}\boldsymbol{U}^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{A}}\right) + o(\boldsymbol{\varepsilon}) \\ &= L(\boldsymbol{A}) + \mu \cdot Trace\left(\left(\frac{\partial L}{\partial \boldsymbol{A}}\right)^{\mathrm{T}} \cdot \boldsymbol{U}\boldsymbol{U}^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{A}}\right) + o(\boldsymbol{\varepsilon}) \\ &= L(\boldsymbol{A}) + \mu \cdot Trace\left(\left(\boldsymbol{U}^T \frac{\partial L}{\partial \boldsymbol{A}}\right)^{\mathrm{T}} \cdot \left(\boldsymbol{U}^T \frac{\partial L}{\partial \boldsymbol{A}}\right)\right) + o(\boldsymbol{\varepsilon}) \\ &= L(\boldsymbol{A}) + \mu \cdot \left\| \boldsymbol{U}^{\mathrm{T}} \frac{\partial L}{\partial \boldsymbol{A}} \right\|_{Fro}^2 + o(\boldsymbol{\varepsilon}), \end{aligned} \tag{A7}$$

where $\|\cdot\|_{Fro}$ is the *Frobenius norm*, and $Trace(\cdot)$ is the *trace* of a matrix. From (A7), if $\mu > 0$ is sufficiently small, we have $L(\boldsymbol{A}+\boldsymbol{\varepsilon}) \geqslant L(\boldsymbol{A})$ because $\left\| \boldsymbol{U}^T \frac{\partial L}{\partial \boldsymbol{A}} \right\|_{Fro} \geqslant 0$. Thus, the conclusion $L(\boldsymbol{A}^{(k+1)}) = L(\boldsymbol{A}^{(k)} + \mu \nabla L(\boldsymbol{A}^{(k)})) \geqslant L(\boldsymbol{A}^{(k)})$ is proven.

Remark. Especially, if $\boldsymbol{U} \neq \boldsymbol{0}, \frac{\partial L}{\partial \boldsymbol{A}} \neq \boldsymbol{0}$, we have $L(\boldsymbol{A}^{(k+1)}) > L(\boldsymbol{A}^{(k)})$.

1 Cao X R, Liu R W. General approach to blind source separation. IEEE Trans Sig Proc, 1996, 44: 562－571

2 Comon P. Independent component analysis, a new concept? Sig Proc, 1992, 36(3): 287－314

3 Suzuki K, Kiryu T, Nakada T. Fast and precise independent component analysis for high field fMRI time series tailored using prior information on spatiotemporal structure. Human Brain Map, 2001, 15: 54－66

4 Lu W, Rajapakse J C. Constrained ICA. NIPS, 2000, 570－576

5 Lu W, Rajapakse JC. ICA with reference. Proc ICA2001, 2001, 120－125

6 Bell A J, Sejnowski T J. An information-maximization approach to blind source separation and blind deconvolution. Neural Comput, 1995, 7: 1129－1159
7 Xie S L, Zhang J L. Blind separation algorithm of minimal mutual information based on rotating transform. Acta Elect Sin (in Chinese), 2002, 30(5): 628－631
8 Hyvarinen A, Oja E. A fast fixed-point algorithm for independent component analysis. Neural Comput, 1997, 9(7): 1483－1492
9 Tan H Z, Chow T W S. Blind and total identification of ARMA models in higher order cumulants domain. IEEE Trans Indust Elect, 1999, 46(6): 1233－1240
10 Tan H Z, Chow T W S. Blind identification of quadratic nonlinear models using neural networks with higher order cumulants. IEEE Trans Indust Elect, 2000, 47(3): 687－696
11 Tan H Z, Aboulnasr T. TOM-based blind identification of nonlinear Volterra systems. IEEE Trans Instr Meas, 2006, 55(1): 300－310
12 Hyvarinen A. Blind source separation by nonstationarity of variance: a cumulant-based approach. IEEE Trans Neural Network, 2001, 12(6): 1471－1474
13 Matsuoka K, Ohya M, Kawamoto M. A neural net for blind separation of nonstationary signals. Neural Networks, 1995, 8(3): 411－419
14 Zhang J L, He Z S, Xie S L. Geometric blind separation algorithm for many source signals. Chinese J Comput (in Chinese), 2005, 28(9): 1575－1581
15 Li Y Q, Wang J. Blind extraction of singularly mixed source signals. IEEE Trans Sign Proc, 2000, 11: 1413－1422
16 Li Y Q, Wang J. Sequential blind extraction of instantaneously mixed sources. IEEE Trans Sig Proc, 2002, 50(5): 997－1006
17 Zibulevsky M, Pearlmutter B A. Blind source separation by sparse decomposition in a signal dictionary. Neural Comput, 2001, 13(4): 863－882
18 Belouchrani A, Cardoso J F. Maximum likelihood source separation for discrete sources. Proc EUSIPCO, 1994, 768－771
19 Lee T W, Lewicki M S, Girolami M, et al. Blind source separation of more sources than mixtures using overcomplete representation. IEEE Sig Proc Lett, 1999, 6(4): 87－90
20 Lewicki M S, Sejnowski T J. Learning overcomplete representations. Neural Comput, 2000, 12: 337－365
21 Li Y Q, Cichocki A, Amari S. Analysis of sparse representation and blind source separation. Neural Comput, 2004, 16(6): 1193－1234
22 Bofill P, Zibulevsky M. Underdetermined source separation using sparse representations. Sig Proc, 2001, 81: 2353－2362
23 Lee T W, Lewicki M, Sejnowski T. ICA mixture models for unsupervised classification of nongaussian sources and automatic context switching in blind signal separation. IEEE Trans Patt Recogn Mach Intel, 2000, 22(10): 1－12
24 Zhang L Q, Amari S, Cichocki A. Natural gradient approach to blind separation of over- and undercomplete mixtures. In: Proc of ICA1999, Aussois, France, Jan. 1999. 455－460
25 Donoho D L, Elad M. Maximal sparsity representation via l^1 minimization. In: Proc National Academy Science, 2003, 100: 2197－2202
26 Takigawa I, Kudo M, Toyama J. Performance analysis of minimum l_1-norm solutions for underdetermined source separation. IEEE Trans Sig Proc, 2004, 52(3): 582－591
27 Parra L, Spence C. Convolutive blind separation of nonstationary sources. IEEE Trans Speech Audio Proc, 2000, 8: 320－327