Contributed article

Equivariant nonstationary source separation

Seungjin Choi[a,*], Andrzej Cichocki[b], Shunichi Amari[b]

[a]*Department of Computer Science and Engineering, Pohang University of Science and Technology, San 31 Hyoja-dong, Nam-gu, Pohang 790-784, South Korea*
[b]*Brain-style Information Systems Research Group, Brain Science Institute, Riken, Japan*

Abstract

Most of source separation methods focus on stationary sources, so higher-order statistics is necessary for successful separation, unless sources are temporally correlated. For nonstationary sources, however, it was shown [Neural Networks 8 (1995) 411] that source separation could be achieved by second-order decorrelation. In this paper, we consider the cost function proposed by Matsuoka et al. [Neural Networks 8 (1995) 411] and derive natural gradient learning algorithms for both fully connected recurrent network and feedforward network. Since our algorithms employ the natural gradient method, they possess the equivariant property and find a steepest descent direction unlike the algorithm [Neural Networks 8 (1995) 411]. We also show that our algorithms are always locally stable, regardless of probability distributions of nonstationary sources. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Blind source separation; Decorrelation; Independent component analysis; Natural gradient; Nonstationarity

1. Introduction

Source separation has received lots of attractions in signal processing and neural network communities because it is a fundamental problem encountered in many applications, such as telecommunications, image/speech processing, and biomedical signals analysis, etc. The task of source separation is to estimate the mixing matrix or to recover original sources, given only their mixtures.

The key assumption in source separation lies in the statistical independence of sources. When sources are mutually independent and are also temporally i.i.d. nonGaussian signals, it is necessary to use higher-order statistics (HOS) to achieve source separation. In such a case, source separation is closely related to independent component analysis (ICA), the goal of which is to decompose multivariate data into a linear sum of nonorthogonal basis vectors with basis coefficients being statistically independent. Along this line many source separation methods have been developed (for example, see Haykin (2000) and Hyvärinen, Karhunen, and Oja (2001) and references therein). In those methods stationary sources were considered, so HOS was necessary implicitly or explicitly.

When sources are spatially uncorrelated but are temporally correlated, we can incorporate the temporal structure of sources into source separation. In such a case, only second-order statistics (SOS) is sufficient for source separation (for example, see Attias and Schreiner (1998), Belouchrani, Abed-Merain, Cardoso, and Moulines (1997) and Pearlmutter and Parra (1997) and references therein).

Another SOS-based source separation method is to exploit the nonstationarity of sources. The nonstationarity of sources was first exploited by Matsuoka et al. in the context of source separation (Matsuoka, Ohya, & Kawamoto, 1995). Some recent work on nonstationary source separation can also be found in Choi and Cichocki (2000a,b), Choi, Cichochi, and Belouchrani (2001) and Pham and Cardoso (2000). Since many natural signals inherently are nonstationary (in the sense that their variances are time varying), it might be useful to take it into account for source separation. In Matsuoka et al. (1995), a linear feedback neural network (see Fig. 1) was employed and a simple learning algorithm was derived using the gradient descent method with a certain approximation. The approximation made in Matsuoka et al. (1995) to simplify the learning algorithm, might be reasonable for the case of two sources ($n = 2$), however, it is questionable for general case, $n \geq 3$. In this paper, we consider a fully connected recurrent network and a feedforward network and derive new source separation algorithms using the natural gradient method developed by Amari (1998). Since the natural gradient method was shown to be efficient

* Corresponding author. Tel.: +82-54-279-2259; fax: +82-54-279-2299.
E-mail addresses: seungjin@postech.ac.kr (S. Choi), cia@brain.riken.go.jp (A. Cichocki), amari@brain.riken.go.jp (S. Amari).
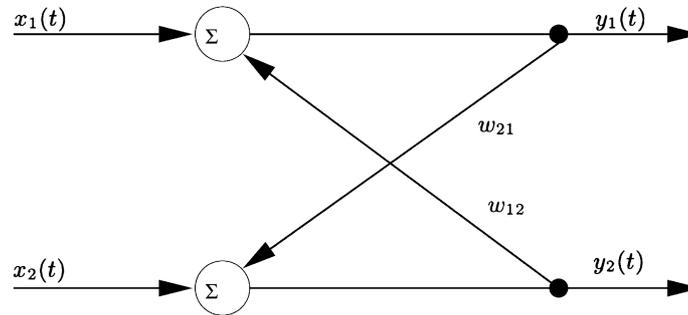
Fig. 1. A recurrent network.

in on-line learning (Amari, 1998), the resulting algorithms are also efficient. In addition, our algorithms enjoy the equivariant property (uniform performance regardless of the mixing condition) due to the nature of the natural gradient. We have to point out that the equivariant property in source separation was observed by Cichocki and Unbehauen (1996) and was elucidated by Cardoso and Laheld from the viewpoint of the relative gradient (Cardoso & Laheld, 1996).

The rest of the paper is organized as follows. The problem formulation and model assumptions are presented in Section 2. In Section 3, the cost function proposed by Matsuoka et al. is briefly reviewed for our further development. In Section 4, we present new efficient source separation algorithms in the framework of the natural gradient. Local stability analysis of the algorithms is given in Section 5. Computer simulation results are presented in Section 6. Conclusions with some discussions are drawn in Section 7.

2. Problem formulation and model assumptions

2.1. Problem formulation

Let us assume that the m-dimensional vector of sensor signals, $\mathbf{x}(t) = [x_1(t), ..., x_m(t)]^\mathrm{T}$ is generated by a linear generative model

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \tag{1}$$

where $\mathbf{s}(t) = [s_1(t), ..., s_n(t)]^\mathrm{T}$ is the n-dimensional vector whose elements are called sources. The matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ is called a mixing matrix. It is assumed that source signals $\{s_i(t)\}$ are statistically independent. The number of sensors, m is greater than or equal to the number of sources, n.

The task of source separation is to recover source vector $\mathbf{s}(t)$ from the observation vector $\mathbf{x}(t)$ without the knowledge of $\mathbf{A}$ or $\mathbf{s}(t)$. In other words, source separation aims at finding a linear mapping (recognition model) which transforms sensor signals $\{x_i(t)\}$ to the output signals $\{y_i(t)\}$, such that the signals $\{y_i(t)\}$ are possibly rescaled estimates of sources $\{s_i(t)\}$. Due to the lack of prior information, there are two indeterminacies in source separation (Comon, 1994): (1) scaling ambiguity and (2) permutation ambiguity.

Thus, the possible estimates of sources, $\mathbf{y}(t)$, that is the output of the demixing filter $\mathbf{W}$ have the relation

$$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t) = \mathbf{W}\mathbf{A}\mathbf{s}(t) = \mathbf{P}\mathbf{D}\mathbf{s}(t), \tag{2}$$

where $\mathbf{P}$ is some permutation matrix and $\mathbf{D}$ is some non-singular diagonal matrix.

2.2. Model assumptions

As in Matsuoka et al. (1995), the following assumptions are made throughout this paper:

AS1: The mixing matrix $\mathbf{A}$ has full column rank.
AS2: Source signals $\{s_i(t)\}$ are statistically independent with zero mean. This implies that the covariance matrix of source signal vector, $\mathbf{R}_s(t) = E\{\mathbf{s}(t)\mathbf{s}^\mathrm{T}(t)\}$ is a diagonal matrix, i.e.

$$\mathbf{R}_s(t) = \mathrm{diag}\{r_1(t), ..., r_n(t)\}, \tag{3}$$

where $r_i(t) = E\{s_i^2(t)\}$ and E denotes the statistical expectation operator.
AS3: $r_i(t)/r_j(t)$ $(i, j = 1, ..., n$ and $i \neq j)$ are not constant with time.

We have to point out that the first two assumptions (AS1 and AS2) are common in most existing approaches to source separation, however, the third assumption (AS3) is critical in the present paper. For nonstationary sources, the third assumption is satisfied and it allows us to separate linear mixtures of sources via SOS.

3. Cost function

For stationary source separation, the typical cost function is based on the mutual information which requires the knowledge of underlying distributions of sources. Since probability distributions of sources are not known in advance, most ICA algorithms rely on hypothesized distributions (for example, see Choi, Cichocki, and Amari (2000) and references therein). HOS should be incorporated either explicitly or implicitly.

For nonstationary sources, Matsuoka et al. have shown that the decomposition (2) is satisfied if cross-correlations

$E\{y_i(t)y_j(t)\}$ $(i,j = 1,...,n, \; i \neq j)$ are zeros at any time instant t, provided that the assumptions (AS1)–(AS3) are satisfied. To eliminate cross-correlations, the following cost function was proposed in Matsuoka et al. (1995)

$$\mathscr{J}(\mathbf{W}) = \frac{1}{2}\left\{\sum_{i=1}^{n} \log E\left\{y_i^2(t)\right\} - \log \det(E\{\mathbf{y}(t)\mathbf{y}^{\mathrm{T}}(t)\})\right\}, \quad (4)$$

where $\det(\cdot)$ denotes the determinant of a matrix. The cost function given in Eq. (4) is a nonnegative function which takes minima if and only if $E\{y_i(t)y_j(t)\} = 0$, for $(i,j = 1,...,n, \; i \neq j)$. This is the direct consequence of the Hadamard's inequality which is summarized below.

Theorem 1 (Hadamard's inequality). *Suppose* $\mathbf{K} = [k_{ij}]$ *is a nonnegative definite symmetric* $n \times n$ *matrix. Then*

$$\det(\mathbf{K}) \leq \prod_{i=1}^{n} k_{ii}, \quad (5)$$

with equality iff $k\{ij\} = 0$, *for* $i \neq j$.

Take the logarithm on both sides of Eq. (5) to obtain

$$\sum_{i=1}^{n} \log k_{ii} - \log \det(\mathbf{K}) \geq 0. \quad (6)$$

Replacing the matrix $\mathbf{K}$ by $E\{\mathbf{y}(t)\mathbf{y}^{\mathrm{T}}(t)\}$, one can easily see that the cost function (4) has the minima iff $E\{y_i(t)y_j(t)\} = 0$, for $i,j = 1,...,n$ and $i \neq j$.

4. Learning algorithms

We first briefly review the learning algorithm that was proposed by Matsuoka et al. Then, we derive new learning algorithms for both fully connected recurrent network and feedforward network. For the sake of simplicity, we only consider the case where there are as many sensors as sources, i.e. $m = n$.

4.1. Matsuoka–Ohya–Kawamoto algorithm

Matsuoka et al. (1995) considered the recurrent network as shown in Fig. 1. For the case of $n = 2$, the input–output description of the network is given by

$$y_1(t) = x_1(t) + w_{12}y_2(t), \qquad y_2(t) = x_2(t) + w_{21}y_1(t). \quad (7)$$

The gradient of the cost function (4) is given by

$$\frac{\mathrm{d}\mathscr{J}(\mathbf{W})}{\mathrm{d}w_{12}} = \frac{1}{1 - w_{12}w_{21}}\frac{E\{y_1(t)y_2(t)\}}{E\{y_1^2(t)\}}, \quad (8)$$

$$\frac{\mathrm{d}\mathscr{J}(\mathbf{W})}{\mathrm{d}w_{21}} = \frac{1}{1 - w_{12}w_{21}}\frac{E\{y_2(t)y_1(t)\}}{E\{y_2^2(t)\}}. \quad (9)$$

In Matsuoka et al. (1995), a simplification was made by eliminating the common term $1/(1 - w_{12}w_{21})$ in Eqs. (8) and (9). While taking this simplification into account and using the stochastic approximation, the learning algorithm is given by

$$w_{12}(t + 1) = w_{12}(t) - \eta_t \frac{y_1(t)y_2(t)}{\lambda_1(t)},$$

$$w_{21}(t + 1) = w_{21}(t) - \eta_t \frac{y_2(t)y_1(t)}{\lambda_2(t)}, \quad (10)$$

where $\eta_t > 0$ is a learning rate and $\lambda_i(t) = E\{y_i^2(t)\}$, is the variance of $y_i(t)$ that is estimated via

$$\lambda_i(t) = (1 - \delta)\lambda_i(t - 1) + \delta y_i^2(t), \quad (11)$$

for some small δ (say, $\delta = 0.01$).

For the case of $(n \geq 3)$, the gradient descent method for the minimization of the cost function (4) leads to the algorithm that has the form

$$\mathbf{W}(t + 1) = \mathbf{W}(t) + \eta_t\{\mathbf{I} - \mathbf{W}(t)\}^{-1}\{\mathbf{I} - \mathbf{\Lambda}^{-1}(t)\mathbf{y}(t)\mathbf{y}^T(t)\}, \quad (12)$$

where all the diagonal elements of $\mathbf{W}(t) = [w_{ij}(t)]$ are zeros. The matrix $\mathbf{\Lambda}(t)$ is the diagonal matrix whose diagonal elements are $\{\lambda_i(t)\}$. Then, the algorithm (10) can be viewed as a simplified version of the algorithm (12) by removing the term $(\mathbf{I} - \mathbf{W})^{-1}$. The factor $(\mathbf{I} - \mathbf{W})^{-1}$ is a common term for the case of $n = 2$ (that was shown earlier), but it is not for general case. Thus, it is questionable whether the same simplification (the elimination of $(\mathbf{I} - \mathbf{W})^{-1}$) is possible for $n \geq 3$. There is no mathematical justification for this simplification. In fact, in Section 6, we will demonstrate that this simplification results in severe degradation or even failure.

4.2. Natural gradient-based algorithms

Gradient descent learning is a popular method to derive a learning algorithm for the purpose of minimizing a given cost function. When a parameter space (on which a cost function is defined) is a Euclidean space with an orthogonal coordinate system, the conventional gradient gives the steepest descent direction. However, if a parameter space is a curved manifold (Riemannian space), an orthonormal linear coordinate system does not exist and the conventional gradient does not give the steepest descent direction (Amari, 1998). Recently, the *natural gradient* was proposed by Amari (1998) and it was shown to be efficient in on-line learning. See Amari (1998) for more details of natural gradient and related work can be found in Amari, Douglas, Cichocki, and Yang (1997), Choi, Amari, and Cichocki (2000), Choi, Amari, Cichocki, and Liu (1999) and Choi, Cichocki, and Amari (1999).

4.2.1. Fully connected recurrent network

We consider a fully connected recurrent network as shown in Fig. 2 for source separation task. The output of the network, $\mathbf{y}(t)$ is given by

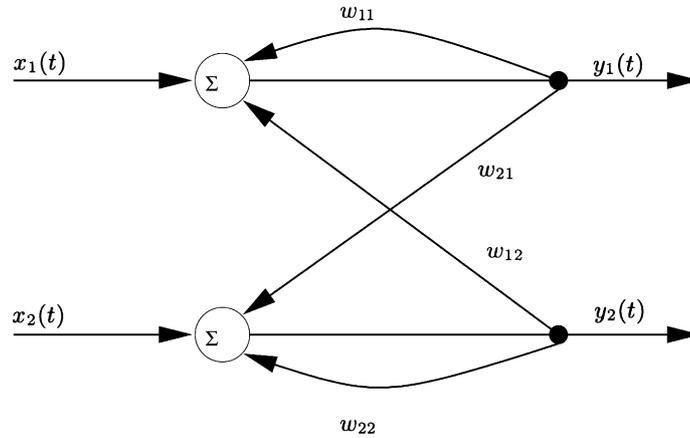$$\mathbf{y}(t) = \mathbf{x}(t) + \mathbf{W}\mathbf{y}(t) = (\mathbf{I} - \mathbf{W})^{-1}\mathbf{x}(t). \quad (13)$$

Fig. 2. A fully connected recurrent network.

We calculate the total differential $d\mathscr{J}(\mathbf{W})$,

$$d\mathscr{J}(\mathbf{W}) = \mathscr{J}(\mathbf{W} + d\mathbf{W}) - \mathscr{J}(\mathbf{W}) = \frac{1}{2}d\left\{\sum_{i=1}^{n}\log E\left\{y_i^2(t)\right\}\right\}$$

$$-\frac{1}{2}d\{\log \det(E\{\mathbf{y}(t)\mathbf{y}^{\mathrm{T}}(t)\})\}, \qquad (14)$$

due to the change $d\mathbf{W}$. Let us define

$$\mathbf{C}(t) = E\{\mathbf{x}(t)\mathbf{x}^{\mathrm{T}}(t)\}. \qquad (15)$$

Then, we have

$$d\{\log \det(E\{\mathbf{y}(t)\mathbf{y}^{\mathrm{T}}(t)\})\} = 2d\{\log \det(\mathbf{I} - \mathbf{W})^{-1}\}$$

$$+ d\{\log \det \mathbf{C}(t)\} = 2\mathrm{tr}\{(\mathbf{I} - \mathbf{W})^{-1} d\mathbf{W} \qquad (16)$$

$$+ d\{\log \det \mathbf{C}(t)\},$$

where $\mathrm{tr}\{\cdot\}$ denotes the trace. Note that the term $\mathbf{C}(t)$ does not depend on the weight matrix $\mathbf{W}$ so it can be eliminated.

Define a modified differential matrix $d\mathbf{V}$ as

$$d\mathbf{V} = (\mathbf{I} - \mathbf{W})^{-1} d\mathbf{W}. \qquad (17)$$

Then

$$d\{\log \det(E\{\mathbf{y}(t)\mathbf{y}^{\mathrm{T}}(t)\})\} = 2\mathrm{tr}\{d\mathbf{V}\}. \qquad (18)$$

Similarly, we have

$$d\left\{\sum_{i=1}^{n}\log E\left\{y_i^2(t)\right\}\right\} = \sum_{i=1}^{n}\frac{2E\{y_i(t)dy_i(t)\}}{E\{y_i^2(t)\}}$$

$$= 2E\{\mathbf{y}^{\mathrm{T}}(t)\mathbf{\Lambda}^{-1}(t)d\mathbf{y}(t)\} = 2E\{\mathbf{y}^{\mathrm{T}}(t)\mathbf{\Lambda}^{-1}(t)d\mathbf{V}\mathbf{y}(t)\}, \ (19)$$

where $\mathbf{\Lambda}(t)$ is a diagonal matrix whose ith diagonal element is $E\{y_i^2(t)\}$.

In terms of $d\mathbf{V}$, we have

$$\frac{d\mathscr{J}(\mathbf{W})}{d\mathbf{V}} = E\{\mathbf{\Lambda}^{-1}(t)\dot{\mathbf{y}}(t)\mathbf{y}^{\mathrm{T}}(t)\} - \mathbf{I}. \qquad (20)$$

By stochastic gradient descent method, the on-line learning

algorithm for updating $\mathbf{V}$ is given by

$$\mathbf{V}(t + 1) = \mathbf{V}(t) + \eta_t\{\mathbf{I} - \mathbf{\Lambda}^{-1}(t)\mathbf{y}(t)\mathbf{y}^{\mathrm{T}}(t)\}, \qquad (21)$$

where $\mathbf{\Lambda}(t)$ is a diagonal matrix whose ith diagonal element is $\lambda_i(t)$ that can be estimated by Eq. (11). Note that $\Delta\mathbf{V}(t) = \mathbf{V}(t + 1) - \mathbf{V}(t)$ has the following relation:

$$\Delta\mathbf{V}(t) = \{\mathbf{I} - \mathbf{W}(t)\}^{-1}\Delta\mathbf{W}(t), \qquad (22)$$

where $\Delta\mathbf{W}(t) = \mathbf{W}(t + 1) - \mathbf{W}(t)$. Thus, the learning algorithm for $\mathbf{W}$ is given by

$$\Delta\mathbf{W}(t) = \eta_t\{\mathbf{I} - \mathbf{W}(t)\}\{\mathbf{I} - \mathbf{\Lambda}^{-1}(t)\mathbf{y}(t)\mathbf{y}^{\mathrm{T}}(t)\}. \qquad (23)$$

Remark.

- It should be noted that the algorithm (23) derived earlier can be viewed as a special form of the robust neural ICA algorithms developed by Cichocki and Unbehauen (1996). In Cichocki and Unbehauen (1996), they considered the fully connected recurrent network and proposed the algorithm that has the form

$$\Delta\mathbf{W}(t) = \eta_t\{\mathbf{I} - \mathbf{W}(t)\}\{\mathbf{I} - f(\mathbf{y}(t))g^{\mathrm{T}}(\mathbf{y}(t))\}, \qquad (24)$$

where $f(\cdot)$ and $g(\cdot)$ are pre-specified element-wise nonlinear functions. The algorithm (23) coincides with algorithm (24) when the nonlinear functions are selected as $f(\mathbf{y}(t)) = \mathbf{\Lambda}^{-1}(t)\mathbf{y}(t)$ and $g(\mathbf{y}(t)) = \mathbf{y}(t)$. However, we arrive at our algorithm (23) using the natural gradient method and a proper cost function that was not exploited in Cichocki and Unbehauen (1996).

- We can rewrite the algorithm (23) as

$$\Delta\mathbf{W}(t) = \eta_t\mathbf{\Lambda}^{-1}(t)\{\mathbf{I} - \mathbf{W}(t)\}\{\mathbf{\Lambda}(t) - \mathbf{y}(t)\mathbf{y}^{\mathrm{T}}(t)\}. \qquad (25)$$

We have to point out that the algorithm (25) leads to a simple form of nonholonomic ICA algorithms proposed by Amari, Chen, and Cichocki (2000) with a variable step size $\eta_t\mathbf{\Lambda}^{-1}(t)$ for nonstationary sources.
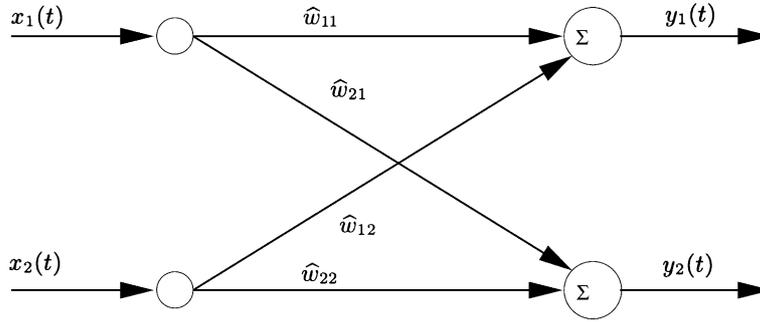
Fig. 3. A fully connected feedforward network.

4.2.2. Fully connected feedforward network

Let us consider a linear feedforward network (see Fig. 3) whose output $\mathbf{y}(t)$ is given by

$$\mathbf{y}(t) = \hat{\mathbf{W}}\mathbf{x}(t). \tag{26}$$

One can easily see that

$$d\{\log \det(E\{\mathbf{y}(t)\mathbf{y}^T(t)\}) = 2d\{\log \det\{\hat{\mathbf{W}}\} + d\{\log \det \mathbf{C}(t)\}$$

$$= 2\text{tr}\{\hat{\mathbf{W}}^{-1} d\hat{\mathbf{W}}\} + d\{\log \det \mathbf{C}(t)\}. \tag{27}$$

Define a modified differential matrix $d\hat{\mathbf{V}}$ as

$$d\hat{\mathbf{V}} = \hat{\mathbf{W}}^{-1} d\hat{\mathbf{W}}. \tag{28}$$

Then, we have

$$d\left\{\sum_{i=1}^{n} \log E\left\{y_i^2(t)\right\}\right\} = 2E\{\mathbf{y}^T(t)\mathbf{\Lambda}^{-1}(t)d\hat{\mathbf{V}}\mathbf{y}(t)\}. \tag{29}$$

Similarly, we can derive the learning algorithm for $\hat{\mathbf{W}}$ that has the form

$$\Delta\hat{\mathbf{W}}(t) = \eta_t\{\mathbf{I} - \mathbf{\Lambda}^{-1}(t)\mathbf{y}(t)\mathbf{y}^T(t)\}\hat{\mathbf{W}}(t)$$

$$= \eta_t\mathbf{\Lambda}^{-1}(t)\{\mathbf{\Lambda}(t) - \mathbf{y}(t)\mathbf{y}^T(t)\}\hat{\mathbf{W}}(t). \tag{30}$$

Note that two remarks described in the case of the fully connected recurrent network also hold in this case (Fig. 3).

5. Local stability analysis

In nonstationary source separation algorithms (23) and (30), the stationary points satisfy

$$E\{\mathbf{I} - \mathbf{\Lambda}^{-1}(t)\mathbf{y}(t)\mathbf{y}^T(t)\} = 0. \tag{31}$$

It follows from Eq. (31) that the stationary points of Eq. (23) or (30) occur if $E\{y_i(t)y_j(t)\} = 0$ for $i, j = 1, \ldots, n$, $i \neq j$. Hence nonstationary source separation is achieved by Eq. (23) or (30). We investigate the local stability of the stationary points of algorithms (23) and (30). In fact, the gradient of the cost function (4) with respect to modified differential matrix $d\mathbf{V}$ (in the recurrent network) or $d\hat{\mathbf{V}}$ (in the feedforward network) is identical, the same stability conditions hold for both recurrent and feedforward networks. Here, we investigate the local stability of the algorithm (23).

Let us consider the associated ODE of Eq. (23)

$$\dot{\mathbf{W}}(t) = \eta_t E\{(\mathbf{I} - \mathbf{W}(t))(\mathbf{I} - \mathbf{\Lambda}^{-1}(t)\mathbf{y}(t)\mathbf{y}^T(t))\}, \tag{32}$$

where $\dot{\mathbf{W}}(t) = d\mathbf{W}(t)/dt$. The linearization of Eq. (32) at stationary point results in the variational equation

$$\delta\dot{\mathbf{W}}(t) = \eta_t \frac{\partial[E\{(\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{\Lambda}^{-1}(t)\mathbf{y}(t)\mathbf{y}^T(t))\}]}{\partial\mathbf{W}} \delta\mathbf{W}(t). \tag{33}$$

It follows from Eq. (33) that the stationary points of the algorithm (23) are asymptotically stable if the real parts of all the eigenvalues of the operator $(\partial[E\{(\mathbf{I} - \mathbf{W}) \times (\mathbf{I} - \mathbf{\Lambda}^{-1}(t)\mathbf{y}(t)\mathbf{y}^T(t))\}])/\partial\mathbf{W}$ are negative.

Amari, Chen, and Cichocki (1997) showed that the evaluation of all the eigenvalues of the operator can be done in terms of the modified differential matrix $d\mathbf{V}$. Since the algorithm is derived from the minimization of the cost function (4), the stationary points of Eq. (23) are stable if the Hessian $d^2\mathcal{J}$ is positive. We calculate the Hessian $d^2\mathcal{J}$ in terms of the modified differential matrix $d\mathbf{V}$ following the suggestion in Amari et al. (1997).

For shorthand notation, we omit the time index t in the following analysis. Recall that

$$d\mathcal{J} = E\{\mathbf{y}^T\mathbf{\Lambda}^{-1} d\mathbf{V}\mathbf{y}\} - \text{tr}\{d\mathbf{V}\}. \tag{34}$$

Then, the Hessian $d^2\mathcal{J}$ is

$$d^2\mathcal{J} = E\{\mathbf{y}^T d\mathbf{V}^T\mathbf{\Lambda}^{-1} d\mathbf{y} + \mathbf{y}^T\mathbf{\Lambda}^{-1} d\mathbf{V} d\mathbf{y}\}$$

$$= E\{\mathbf{y}^T d\mathbf{V}^T\mathbf{\Lambda}^{-1} d\mathbf{V}\mathbf{y} + \mathbf{y}^T\mathbf{\Lambda}^{-1} d\mathbf{V} d\mathbf{V}\mathbf{y}\}. \tag{35}$$

The first term of Eq. (35) is

$$E\{\mathbf{y}^T d\mathbf{V}^T\mathbf{\Lambda}^{-1} d\mathbf{y}\} = \sum_{i,j,k} E\left\{y_i \frac{dv_{ji}}{\lambda_j} dv_{jk}y_k\right\} = \sum_{i,j} \frac{\lambda_i}{\lambda_j}(dv_{ji})^2. \tag{36}$$

The second term of Eq. (35) is

$$E\{\mathbf{y}^{\mathrm{T}}\mathbf{\Lambda}^{-1}\,\mathrm{d}\mathbf{V}\,\mathrm{d}\mathbf{y}\} = \sum_{i,j,k} E\left\{\frac{y_i}{\lambda_i}\mathrm{d}v_{ij}\,\mathrm{d}v_{jk}y_k\right\} = \sum_{i,j}\mathrm{d}v_{ij}\,\mathrm{d}v_{ji}.$$
(37)

Note that the statistical expectation is taken at the solution which satisfies the condition $E\{y_iy_j\} = 0$ for $i \neq j$. From Eqs. (36) and (37), we have

$$\mathrm{d}^2\mathscr{J} = \sum_{i,j}\left[\frac{\lambda_i}{\lambda_j}(\mathrm{d}v_{ji})^2 + \mathrm{d}v_{ij}\,\mathrm{d}v_{ji}\right].$$
(38)

Rewrite Eq. (38) as

$$\mathrm{d}^2\mathscr{J} = \sum_{i\neq j}q_{ij} + \sum_i q_{ii},$$
(39)

where

$$q_{ij} = \frac{\lambda_i}{\lambda_j}(\mathrm{d}v_{ji})^2 + \mathrm{d}v_{ij}\,\mathrm{d}v_{ji}.$$
(40)

One can easily see that the summand in the second term in Eq. (39) is always positive. For a pair (i,j), $i \neq j$, the summand in the first term in Eq. (39) can be rewritten as

$$q_{ij} + q_{ji} = \frac{\lambda_i}{\lambda_j}(\mathrm{d}v_{ji})^2 + \frac{\lambda_j}{\lambda_i}(\mathrm{d}v_{ij})^2 + 2\mathrm{d}v_{ij}\,\mathrm{d}v_{ji}$$

$$= [\mathrm{d}v_{ij}\,\mathrm{d}v_{ji}]\begin{bmatrix}\dfrac{\lambda_j}{\lambda_i} & 1 \\ 1 & \dfrac{\lambda_i}{\lambda_j}\end{bmatrix}\begin{bmatrix}\mathrm{d}v_{ij} \\ \mathrm{d}v_{ji}\end{bmatrix}.$$
(41)

One can easily see that $q_{ij} + q_{ji}$ is always nonnegative. Hence $\mathrm{d}^2\mathscr{J}$ is always positive. Note that the stability of the algorithm (23) does not depend on the probability distributions of sources. Thus, our algorithm is always locally stable regardless of the probability distributions of sources. The same stability conditions hold for the algorithm (30).

6. Computer simulation results

We have performed experiments with digitized voice signals sampled at 8 kHz which are shown in Fig. 4. The algorithms (10), (23), and (30) were tested and their performance was compared. All the experiments were carried out with artificial mixing and real world signals. In this framework, we can measure the performance of the
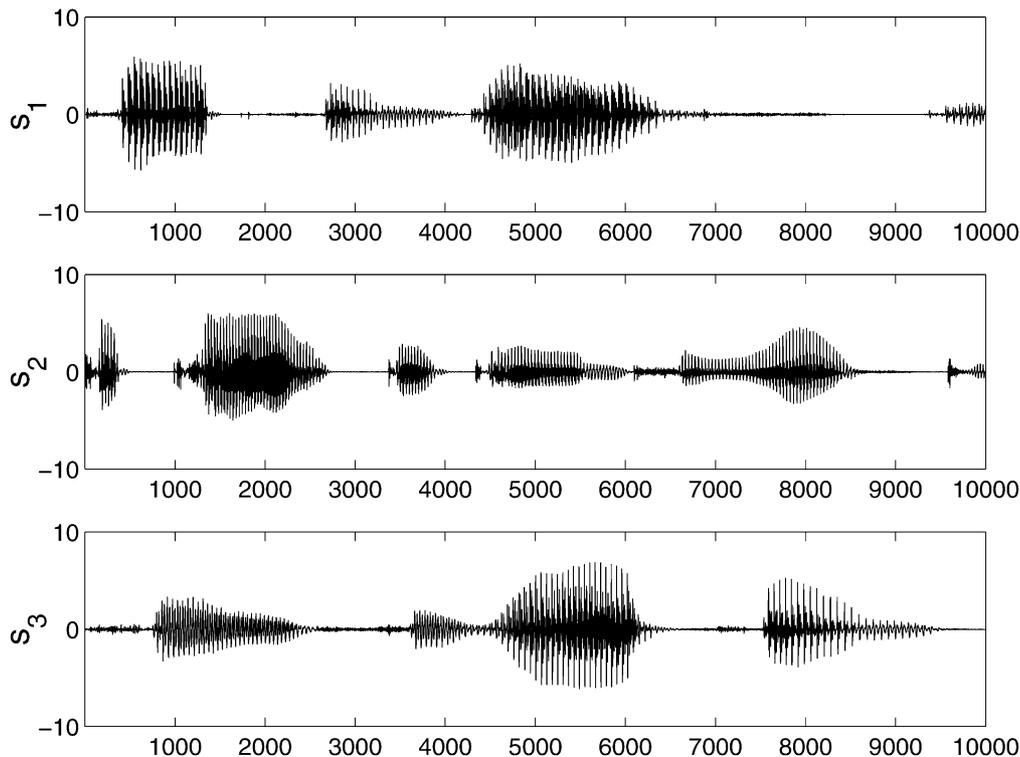

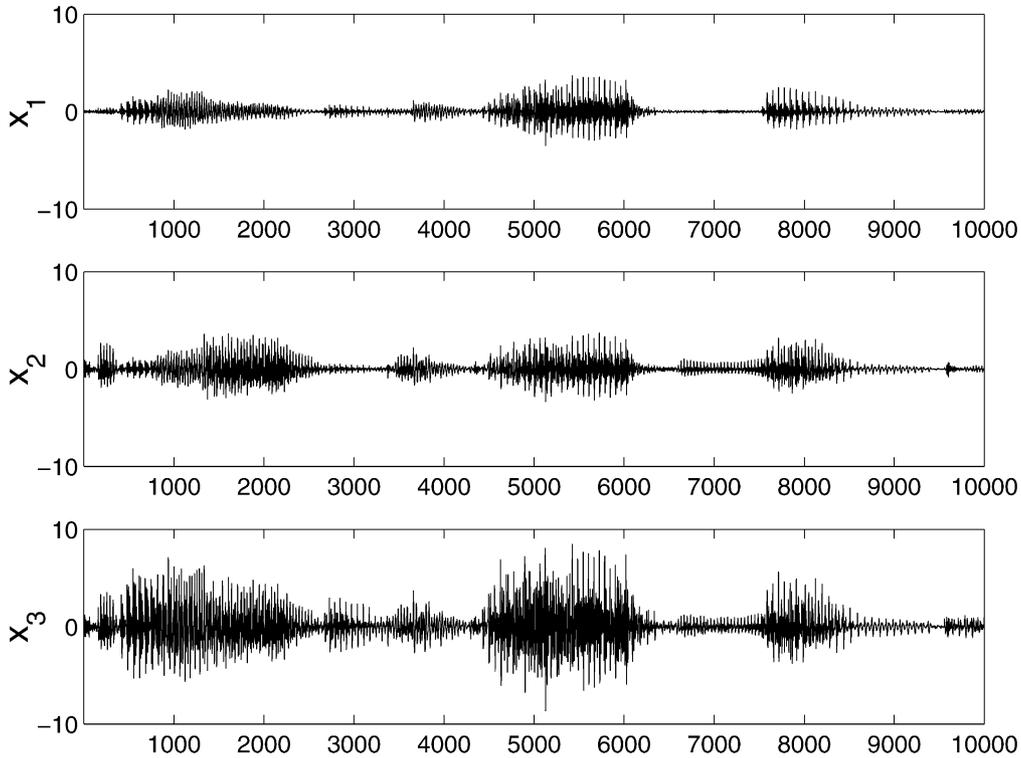Fig. 4. Original voice signals sampled at 8 kHz, $s_1(t)$, $s_2(t)$, and $s_3(t)$ for the duration of 10 000 samples.

Fig. 5. Three mixture signals $x_1(t)$, $x_2(t)$, and $x_3(t)$ in Simulation 2.

algorithm in terms of the performance index (PI) defined by

$$\text{PI} = \sum_{i=1}^{n} \left\{ \left(\sum_{k=1}^{n} \frac{|g_{ik}|}{\max_j |g_{ij}|} - 1 \right) + \left(\sum_{k=1}^{n} \frac{|g_{ki}|}{\max_j |g_{ji}|} - 1 \right) \right\}, \tag{42}$$

where g_{ij} is the (i,j)th element of the global system matrix $\mathbf{G}$ ($\mathbf{G} = (\mathbf{I} - \mathbf{W})^{-1}\mathbf{A}$ for a recurrent network, $\mathbf{G} = \hat{\mathbf{W}}\mathbf{A}$ for a feedforward network) and $\max_j g_{ij}$ represents the maximum value among the elements in the ith row vector of $\mathbf{G}$, $\max_j g_{ji}$ does the maximum value among the elements in the ith column vector of $\mathbf{G}$. The PI defined in Eq. (42) explains us how far the global system matrix $\mathbf{G}$ is from a generalized permutation matrix. When perfect signal separation is achieved, the PI is zero. In practice, when the PI falls below 0.01, the separation is satisfactory.

In addition to the performance measure (42), we also calculated the signal to interference ratio improvement (SIRI) defined by

$$\text{SIRI}_i = \frac{E\{(x_i - s_i)^2\}}{E\{(y_i - s_i)^2\}}. \tag{43}$$

Note that scaling and ordering ambiguities have to be resolved before SIRI is computed. To remove scaling ambiguity, we normalized original voice signals so that they have unit variance. In addition, after the separation was achieved, the recovered signals $\{y_i\}$ were also normalized. Due to the ordering ambiguity, the first original voice signal can be appeared at the second output node of the separation network, i.e. $y_2(t) = s_1(t)$. Or even after normalization, the signal $y_2(t)$ might be the upside-down version of $s_1(t)$, i.e. $y_2(t) = -s_1(t)$. All these things should be taken into account in the calculation of SIRI given in Eq. (43).

In the performance comparison, the algorithms (10), (23), and (30) will be named as Algorithms 1–3, respectively.

6.1. Simulation 1

Our first experiment was performed for $n = 2$. The first two voice signals, $s_1(t)$ and $s_2(t)$, in Fig. 4 were artificially mixed using the mixing matrix $\mathbf{A}$ given by

$$\mathbf{A} = \begin{bmatrix} 1 & 0.7 \\ 0.6 & 1 \end{bmatrix}. \tag{44}$$

This is a well-conditioned mixing (the condition number of $\mathbf{A}$ is 4.7). All three algorithms were successful in separation, their performance was almost similar. The SIRI_1 and SIRI_2 were around 60 dB. This result indicates that the approximation made in Matsuoka et al. (1995) might be reasonable for $n = 2$. However, in the following two computer simulations, we will demonstrate that our proposed algorithms outperform the Algorithm 1 given in Eq. (10) for the case of $n \geq 3$ as well as ill-conditioned mixing.

6.2. Simulation 2

The second experiment was conducted for $n = 3$. Three

Table 1
SIRI in Simulation 2

Type of algorithm	SIRI (dB)
Algorithm 1	$SIRI_1 = 16.0$ $SIRI_2 = 14.8$ $SIRI_3 = 36.5$
Algorithm 2	$SIRI_1 = 68.1$ $SIRI_2 = 61.5$ $SIRI_3 = 65.1$
Algorithm 3	$SIRI_1 = 68.0$ $SIRI_2 = 61.5$ $SIRI_3 = 65.2$

mixture signals were generated using the mixing matrix given by

$$\mathbf{A} = \begin{bmatrix} 0.224 & 0.055 & 0469 \\ 0.162 & 0.505 & 0.476 \\ 0.933 & 0.649 & 0.912 \end{bmatrix}. \tag{45}$$

The initial values of all synaptic weights for the feedback network were set to be zeros and the synaptic weight matrix for the feedforward network was initialized as the identity matrix. The constant learning rate $\eta_t = 0.0005$ was used for all three algorithms (10), (23), and (30). The condition number of the mixing matrix $\mathbf{A}$ given in Eq. (45) is 7.2 (well-conditioned mixing). Three mixture signals, $x_1(t)$, $x_2(t)$, and $x_3(t)$ are shown in Fig. 5. Our proposed algorithms (23) and (30) outperformed the algorithm (10). In terms of

SIRI, the result is summarized in Table 1. Recovered signals using the algorithms (10) and (23) are shown in Figs. 6 and 7, respectively. The separated signals using the algorithm (30) were very close to ones shown in Fig. 7. The evolution of PI is shown in Fig. 8.

6.3. Simulation 3

The third experiment was conducted for the ill-conditioned mixing. The voice signals as shown in Fig. 4 were artificially mixed using the ill-conditioned mixing matrix given by

$$\mathbf{A} = \begin{bmatrix} 1.0 & 0.7 & 0.6 \\ 1.1 & 0.7 & 0.6 \\ 1.3 & 0.5 & 0.3 \end{bmatrix}. \tag{46}$$

In this environment, first voice signal $s_1(t)$ is dominant in all three mixtures. The first and second sensor signals are very close. The condition number of $\mathbf{A}$ is 232.7 (ill-conditioned mixing). The initial conditions and the learning rate were same as the ones in Simulation 2. The algorithm (10) failed to separate mixtures, whereas the algorithms (23) and (30) were still successful. The SIRI was calculated and summarized in Table 2.

7. Conclusions

We have presented two efficient source separation algorithms for the fully connected recurrent network and
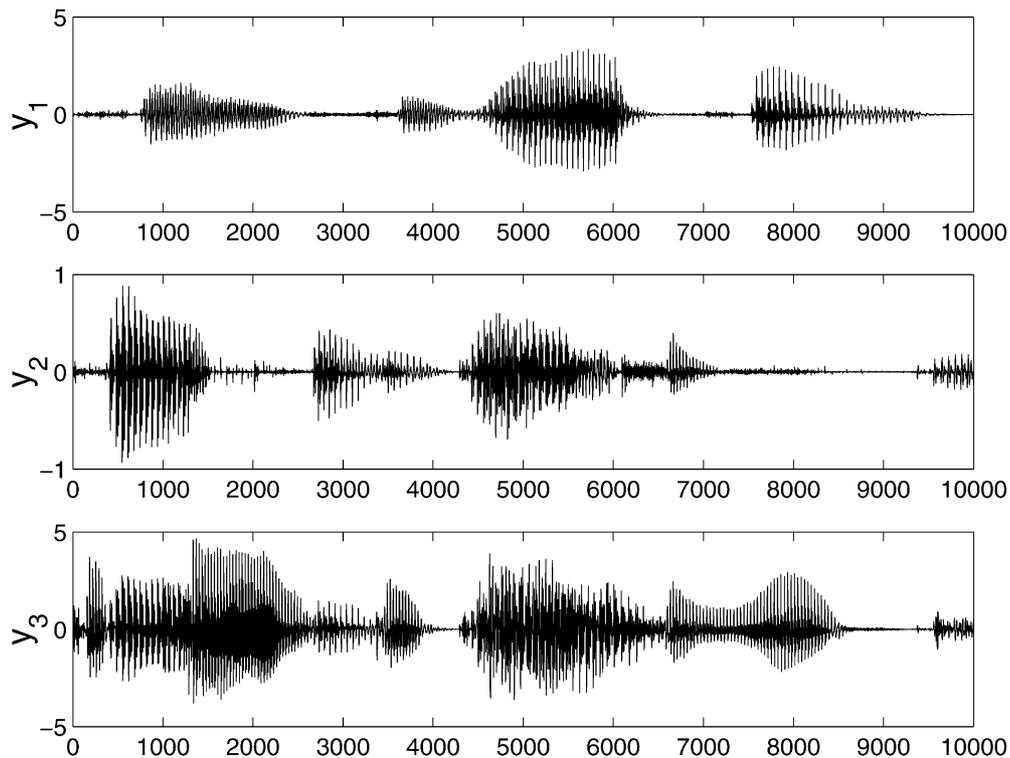

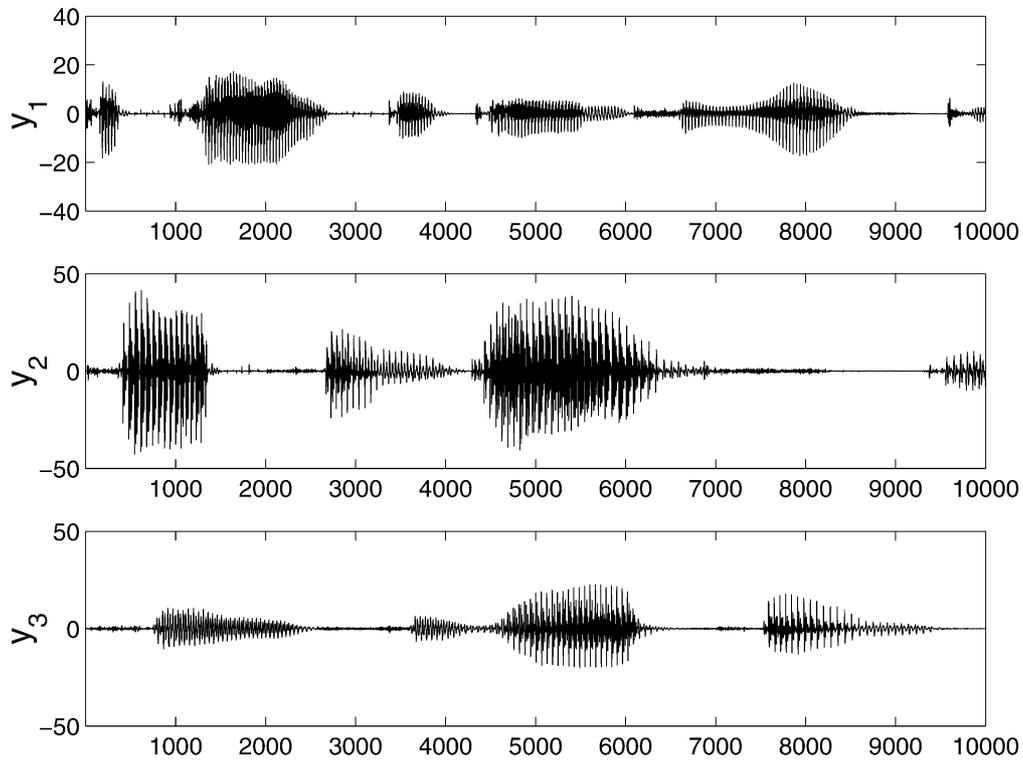Fig. 6. Three recovered signals $y_1(t)$, $y_2(t)$, and $y_3(t)$ using the algorithm (10) in Simulation 2.

Fig. 7. Three recovered signals $y_1(t)$, $y_2(t)$, and $y_3(t)$ using the algorithm (23) in Simulation 2.

feedforward network, when sources are nonstationary. The proposed algorithms were derived in the framework of the natural gradient, hence they are efficient and possess the equivariant property. Rigorous derivation of the algorithms was presented. We also showed that our algorithms were locally stable, regardless of probability distributions of sources. Through computer simulations,

we have demonstrated that the proposed algorithms outperformed the existing algorithm. In this paper, we consider the linear instantaneous mixture case, however, it will be interesting to extend this idea to the convolutive mixture case which has more practical applications. The extension of the proposed method to convolutive mixtures is under investigation.
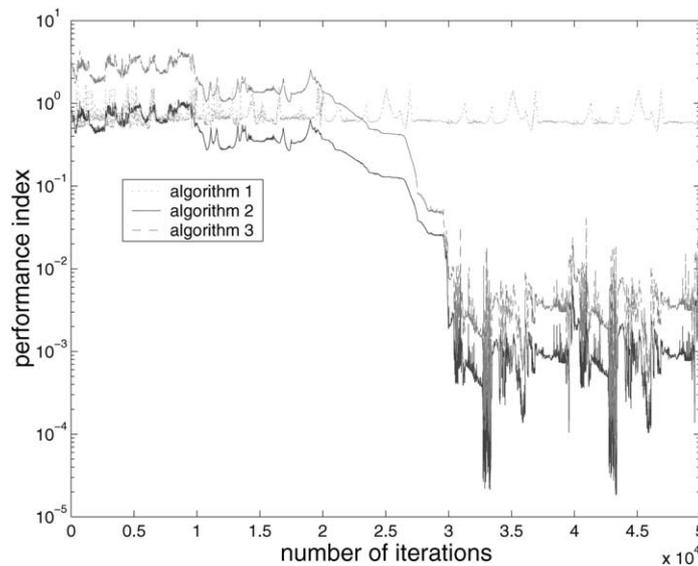

Fig. 8. The evolution of PI in Simulation 2 is shown. Our algorithms (23) and (30) outperform the algorithm (10).

Table 2
SIRI in Simulation 3

Type of algorithm	SIRI (dB)
Algorithm 2	$SIRI_1 = 64.4$
	$SIRI_2 = 63.0$
	$SIRI_3 = 63.9$
Algorithm 3	$SIRI_1 = 64.8$
	$SIRI_2 = 46.4$
	$SIRI_3 = 31.2$

Acknowledgements

References

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, *10* (2), 251–276.

Amari, S., Chen, T. P., & Cichocki, A. (1997). Stability analysis of learning algorithms for blind source separation. *Neural Networks*, *10* (8), 1345–1351.

Amari, S., Douglas, S. C., Cichocki, A., & Yang, H. H. (1997). Multichannel blind deconvolution and equalization using the natural gradient. *Proceedings on SPAWC*, Paris, France (pp. 101–104).

Amari, S., Chen, T. P., & Cickocki, A. (2000). Nonholonomic orthogonal learning algorithms for blind source separation. *Neural Computation*, *12* (6), 1463–1484.

Attias, H., & Schreiner, C. E. (1998). Blind source separation and deconvolution: The dynamic component analysis algorithms. *Neural Computation*, *10*, 1373–1424.

Belouchrani, A., Abed-Merain, K., Cardoso, J. -F., & Moulines, E. (1997). A blind source separation technique using second order statistics. *IEEE Transactions on Signal Processing*, *45*, 434–444.

Cardoso, J. -F., & Laheld, B. H. (1996). Equivariant adaptive source separation. *IEEE Transactions on Signal Processing*, *44* (12), 3017–3030.

Choi, S., & Cichocki, A. (2000a). Blind separation of nonstationary and temporally correlated sources from noisy mixtures. *Proceedings on IEEE Workshop on Neural Networks for Signal Processing*, Sidney, Australia (pp. 405–414).

Choi, S., & Cichocki, A. (2000b). Blind separation of nonstationary sources in noisy mixtures. *Electronics Letters*, *36* (9), 848–849.

Choi, S., Amari, A., Cichocki, A., & Liu, R. (1999). Natural gradient learning with a nonholonomic constraint for blind deconvolution of multiple channels. *Proceedings on ICA'99*, Aussois, France (pp. 371–376).

Choi, S., Cichocki, A., & Amari, S. (1999). Two spatio-temporal decorrelation algorithms and their application to blind deconvolution of multiple channels. *Proceedings on ICASSP*, Phoenix, Arizona (pp. 1085–1088).

Choi, S., Amari, S., & Cichocki, A. (2000). Natural gradient learning for spatio-temporal decorrelation: Recurrent network. *IEICE Transactions on Fundamentals*, *E83-A* (12), 2715–2722.

Choi, S., Cichocki, A., & Amari, S. (2000). Flexible independent component analysis. *Journal of VLSI Signal Processing*, *26* (1/2), 25–38.

Choi, S., Cichocki, A., & Belouchrani, A. (2001). Blind separation of second-order nonstationary and temporally colored sources. *Proceedings on IEEE Workshop on Statistical Signal Processing*, Singapore (pp. 444–447).

Cichocki, A., & Unbehauen, R. (1996). Robust neural networks with on-line learning for blind identification and blind separation of sources. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, *43*, 894–906.

Comon, P. (1994). Independent component analysis, a new concept? *Signal Processing*, *36* (3), 287–314.

Haykin, S. (2000). *Unsupervised adaptive filtering: Blind source separation*, Englewood Cliffs, NJ: Prentice Hall.

Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent component analysis*, New York: Wiley.

Matsuoka, K., Ohya, M., & Kawamoto, M. (1995). A neural net for blind separation of nonstationary signals. *Neural Networks*, *8* (3), 411–419.

Pearlmutter, B., & Parra, L. (1997). Maximum likelihood blind source separation: A context-sensitive generalization of ICA. In M. C. Mozer, M. I. Jordan & T. Petsche, *Advances in neural information processing systems* (pp. 613–619). Vol. 9.

Pham, D. T., & Cardoso, J. -F. (2000). Blind separation of instantaneous mixtures of nonstationary sources. *Proceedings on ICA*, Helsinki, Finland (pp. 187–192).