Natural Gradient Learning for Second-Order Nonstationary Source Separation

Seungjin Choi [§], Andrzej Cichocki [†], Shunichi Amari [†]

[§] Department of Computer Science and Engineering, POSTECH, Korea
seungjin@postech.ac.kr

[†] Brain-style Information Systems Research Group, Brain Science Institute, RIKEN, Japan
{cia,amari}@brain.riken.go.jp

Abstract - In this paper we consider a problem of source separation when sources are second-order nonstationary stochastic processes. We employ the natural gradient method and develop learning algorithms for both linear feedback and feedforward neural networks. Thus our algorithms possess equivariant property. Local stability analysis shows that separating solutions are always locally stable stationary points of the proposed algorithms, regardless of probability distributions of sources.

I. Introduction

In the context of source separation, it is assumed that the m dimensional vector of measurement signals, $\boldsymbol{x}(t)$, is generated by a linear data model described by

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

where $\boldsymbol{s}(t)$ is the n dimensional vector whose elements are called sources. The matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is called a mixing matrix. The task of source separation is to estimate the mixing matrix $\boldsymbol{A}$ (or its inverse which is referred to the demixing matrix), given only a finite number of measurement signals, $\{\boldsymbol{x}(t)\}$, $t = 1, \ldots, N$. Source vector $\boldsymbol{s}(t)$ is unknown in advance, but their elements are assumed to be statistically independent.

Most of source separation methods have focused on stationary sources, so higher-order statistics (HOS) is necessary for successful separation, unless sources are temporally correlated. Along this line, a variety of algorithms have been developed (for example see [1], [2] and references therein).

Many natural signals are inherently nonstationary stochastic processes. Especially they have second-order nonstationary characteristics in the sense that their variances are time-varying. It was shown in [3] that source separation could be achieved by decorrelation if sources are independent second-order nonstationary stochastic processes. Recently some algebraic nonstationary source separation methods have been developed [4], [5], [6], [7], [8].

In this paper we pay our attention to the problem of second-order nonstationary source separation. We develop natural gradient learning algorithms for both linear feedback and feedforward neural networks. Due the natural gradient method, our algorithms converge to separation solutions along the steepest descent direction and possess the equivariant property (uniform performance regardless of mixing condition) that was first discovered by Cardoso and Laheld [9]. We also present local stability analysis of our algorithms and show that separating solutions are always locally stable stationary points of our algorithms, regardless of probability distributions of sources.

As in [3], the following assumptions are made throughout this paper:

AS1 The mixing matrix $\boldsymbol{A}$ 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, $\boldsymbol{R}_s(t) = E\{\boldsymbol{s}(t)\boldsymbol{s}^T(t)\}$ is a diagonal matrix, i.e.,

$$\boldsymbol{R}_s(t) = \mathrm{diag}\{r_1(t), \ldots, r_n(t)\}, \tag{2}$$

where $r_i(t) = E\{s_i^2(t)\}$ and E denotes the statistical expectation operator.

AS3 $\frac{r_i(t)}{r_j(t)}$ $(i, j = 1, \ldots, n \text{ and } i \neq j)$ are not constant with time.

II. Natural Gradient Algorithms

We consider the objective function proposed by Matsuoka *et al.* [3]. Then we employ the natural gradient method which was shown to be efficient for on-line learning [10], [11], [12] and derive on-line learning algorithms for both feedback and feedforward networks. For the sake of simplicity, we only consider the case where there are as many sensors as sources, i.e., $m = n$.

A. Objective Function

It was shown in [3] that second-order decorrelation is sufficient for source separation under the assumptions (AS1)-(AS3). The objective function that we consider is given by

$$\begin{aligned}\mathcal{J}(\boldsymbol{W}) &= \frac{1}{2}\left\{\sum_{i=1}^{n}\log E\{y_i^2(t)\}\right. \\ &\quad \left.-\log\det\left(E\left\{\boldsymbol{y}(t)\boldsymbol{y}^T(t)\right\}\right)\right\}, \end{aligned} \tag{3}$$

where $\boldsymbol{y}(t)$ is the network output vector and $\det(\cdot)$ denotes the determinant of a matrix. The objective function (3) is a non-negative function which takes minima if and only if $E\{y_i(t)y_j(t)\}=0$, for $i,j=1,\ldots,n,\ i\neq j$.

B. Feedback Network

We consider a linear feedback network whose output $\boldsymbol{y}(t)$ is described by

$$\begin{aligned}\boldsymbol{y}(t) &= \boldsymbol{x}(t)+\boldsymbol{W}\boldsymbol{y}(t) \\ &= (\boldsymbol{I}-\boldsymbol{W})^{-1}\boldsymbol{x}(t). \end{aligned} \tag{4}$$

We calculate the total differential $d\mathcal{J}(\boldsymbol{W})$,

$$\begin{aligned}d\mathcal{J}(\boldsymbol{W}) &= \mathcal{J}(\boldsymbol{W}+d\boldsymbol{W})-\mathcal{J}(\boldsymbol{W}) \\ &= \frac{1}{2}d\left\{\sum_{i=1}^{n}\log E\{y_i^2(t)\}\right\} \\ &\quad -\frac{1}{2}d\left\{\log\det\left(E\{\boldsymbol{y}(t)\boldsymbol{y}^T(t)\}\right)\right\}, \end{aligned} \tag{5}$$

due to the change $d\boldsymbol{W}$.

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

$$d\boldsymbol{V}=(\boldsymbol{I}-\boldsymbol{W})^{-1}d\boldsymbol{W}. \tag{6}$$

With this definition, we have

$$\begin{aligned}&d\left\{\log\det(E\{\boldsymbol{y}(t)\boldsymbol{y}^T(t)\})\right\} \\ &= 2\mathrm{tr}\{(\boldsymbol{I}-\boldsymbol{W})^{-1}d\boldsymbol{W}\}+d\left\{\log\det\boldsymbol{C}(t)\right\} \\ &= 2\mathrm{tr}\{d\boldsymbol{V}\}+d\left\{\log\det\boldsymbol{C}(t)\right\}, \end{aligned} \tag{7}$$

where $\mathrm{tr}\{\cdot\}$ denotes the trace operator and $\boldsymbol{C}(t)$ is the covariance matrix of $\boldsymbol{x}(t)$ defined by

$$\boldsymbol{C}(t)=E\{\boldsymbol{x}(t)\boldsymbol{x}^T(t)\}. \tag{8}$$

Note that $\boldsymbol{C}(t)$ does not depend on the weight matrix $\boldsymbol{W}$ so it can be eliminated.

Similarly, we have

$$\begin{aligned}&d\left\{\sum_{i=1}^{n}\log E\{y_i^2(t)\}\right\} \\ &= \sum_{i=1}^{n}\frac{2E\{y_i(t)dy_i(t)\}}{E\{y_i^2(t)\}} \\ &= 2E\{\boldsymbol{y}^T(t)\boldsymbol{\Lambda}^{-1}(t)d\boldsymbol{V}\boldsymbol{y}(t)\}, \end{aligned} \tag{9}$$

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

Hence, the gradient of the objective function (3) with respect to the modified differential matrix $d\boldsymbol{V}$ is given by

$$\frac{d\mathcal{J}(\boldsymbol{W})}{d\boldsymbol{V}}=E\left\{\boldsymbol{\Lambda}^{-1}(t)\boldsymbol{y}(t)\boldsymbol{y}^T(t)\right\}-\boldsymbol{I} \tag{10}$$

The stochastic gradient descent method leads to the updating rule for $\boldsymbol{V}$ that has the form

$$\boldsymbol{V}(t+1)=\boldsymbol{V}(t)+\eta_t\left\{\boldsymbol{I}-\boldsymbol{\Lambda}^{-1}(t)\boldsymbol{y}(t)\boldsymbol{y}^T(t)\right\}, \tag{11}$$

where $\eta_t>0$ is the learning rate and $\boldsymbol{\Lambda}(t)$ is a diagonal matrix whose ith diagonal element is $\lambda_i(t)$ that can be estimated by

$$\lambda_i(t)=(1-\delta)\lambda_i(t-1)+\delta y_i^2(t), \tag{12}$$

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

It follows from the definition (6) that the learning algorithm for $\boldsymbol{W}$ is given by

$$\begin{aligned}\Delta\boldsymbol{W}(t) &= \boldsymbol{W}(t+1)-\boldsymbol{W}(t) \\ &= \eta_t\left\{\boldsymbol{I}-\boldsymbol{W}(t)\right\}\left\{\boldsymbol{I}-\boldsymbol{\Lambda}^{-1}(t)\boldsymbol{y}(t)\boldsymbol{y}^T(t)\right\}. \end{aligned} \tag{13}$$

Remarks

- It should be noted that the algorithm (13) can be viewed as a special form of the robust neural ICA algorithms developed by Cichocki and Unbehauen [13] that has the form

$$\begin{aligned}&\Delta\boldsymbol{W}(t) \\ &= \eta_t\left\{\boldsymbol{I}-\boldsymbol{W}(t)\right\}\left\{\boldsymbol{I}-f(\boldsymbol{y}(t))g^T(\boldsymbol{y}(t))\right\}, \end{aligned} \tag{14}$$

  where $f(\cdot)$ and $g(\cdot)$ are pre-specified element-wise nonlinear functions. The algorithm (13) coincides with (14) when the nonlinear functions are selected as $f(\boldsymbol{y}(t))=\boldsymbol{\Lambda}^{-1}(t)\boldsymbol{y}(t)$ and $g(\boldsymbol{y}(t))=\boldsymbol{y}(t)$. However we arrive at our algorithm (13) using the natural gradient method and a proper objective function that was not exploited in [13].
- We can rewrite the algorithm (13) as

$$\begin{aligned}\Delta\boldsymbol{W}(t) &= \eta_t\boldsymbol{\Lambda}^{-1}(t)\left\{\boldsymbol{I}-\boldsymbol{W}(t)\right\} \\ &\quad \left\{\boldsymbol{\Lambda}(t)-\boldsymbol{y}(t)\boldsymbol{y}^T(t)\right\}. \end{aligned} \tag{15}$$

We have to point out that the algorithm (15) leads to a simple form of nonholonomic ICA algorithms proposed by Amari *et al.* [14] with a variable step size $\eta_t \mathbf{\Lambda}^{-1}(t)$ for nonstationary sources.

C. Feedforward Network

Let us consider a linear feedforward network whose output $\boldsymbol{y}(t)$ is given by

$$\boldsymbol{y}(t) = \widehat{\boldsymbol{W}} \boldsymbol{x}(t). \tag{16}$$

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

$$d\widehat{\boldsymbol{V}} = \widehat{\boldsymbol{W}}^{-1} d\widehat{\boldsymbol{W}}. \tag{17}$$

With this definition, we have

$$\begin{aligned} & d\left\{\log\det(E\{\boldsymbol{y}(t)\boldsymbol{y}^T(t)\})\right\} \\ &= 2\mathrm{tr}\left\{\widehat{\boldsymbol{W}}^{-1} d\widehat{\boldsymbol{W}}\right\} + d\left\{\log\det \boldsymbol{C}(t)\right\}, \end{aligned} \tag{18}$$

and

$$d\left\{\sum_{i=1}^{n} \log E\{y_i^2(t)\}\right\} = 2E\{\boldsymbol{y}^T(t)\mathbf{\Lambda}^{-1}(t) d\widehat{\boldsymbol{V}} \boldsymbol{y}(t)\}. \tag{19}$$

Then, the natural gradient learning algorithm for $\widehat{\boldsymbol{W}}$ has the form

$$\begin{aligned} \Delta\widehat{\boldsymbol{W}}(t) &= \eta_t \left\{\boldsymbol{I} - \mathbf{\Lambda}^{-1}(t)\boldsymbol{y}(t)\boldsymbol{y}^T(t)\right\} \widehat{\boldsymbol{W}}(t) \\ &= \eta_t \mathbf{\Lambda}^{-1}(t) \left\{\mathbf{\Lambda}(t) - \boldsymbol{y}(t)\boldsymbol{y}^T(t)\right\} \widehat{\boldsymbol{W}}(t). \end{aligned} \tag{20}$$

Note that two remarks described for the case of linear feedback network also hold in this case.

III. Local Stability Analysis

Stationary points of the algorithm (13) or (20) satisfy

$$E\left\{\boldsymbol{I} - \mathbf{\Lambda}^{-1}(t)\boldsymbol{y}(t)\boldsymbol{y}^T(t)\right\} = 0, \tag{21}$$

which implies that $E\{y_i(t)y_j(t)\} = 0$ for $i, j = 1, \ldots, n$, $i \neq j$. In order to show that stationary points of (13) are locally stable, we need to show that the Hessian $d^2\mathcal{J}$ is positive. Following suggestions in [15], we calculate the Hessian $d^2\mathcal{J}$ in terms of the modified differential matrix. Note that the gradient of the objective function (3) with respect to modified differential matrix $d\boldsymbol{V}$ (in the feedback network) or $d\widehat{\boldsymbol{V}}$ (in the feedforward network) is identical, the same stability conditions hold for both feedback and feedforward networks. Here we investigate the local stability of the algorithm (13).

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

$$d\mathcal{J} = E\left\{\boldsymbol{y}^T \mathbf{\Lambda}^{-1} d\boldsymbol{V} \boldsymbol{y}\right\} - \mathrm{tr}\left\{d\boldsymbol{V}\right\}. \tag{22}$$

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

$$d^2\mathcal{J} = E\left\{\boldsymbol{y}^T d\boldsymbol{V}^T \mathbf{\Lambda}^{-1} d\boldsymbol{V} \boldsymbol{y} + \boldsymbol{y}^T \mathbf{\Lambda}^{-1} d\boldsymbol{V}\, d\boldsymbol{V} \boldsymbol{y}\right\}. \tag{23}$$

The first term of (23) is

$$E\left\{\boldsymbol{y}^T d\boldsymbol{V}^T \mathbf{\Lambda}^{-1} d\boldsymbol{y}\right\} = \sum_{i,j} \frac{\lambda_i}{\lambda_j} (dv_{ji})^2. \tag{24}$$

The second term of (23) is

$$E\left\{\boldsymbol{y}^T \mathbf{\Lambda}^{-1} d\boldsymbol{V}\, d\boldsymbol{y}\right\} = \sum_{i,j} dv_{ij} dv_{ji}. \tag{25}$$

Note that the statistical expectation is taken at the solution which satisfies the condition $E\{y_i y_j\} = 0$ for $i \neq j$. From (24) and (25), we have

$$d^2\mathcal{J} = \sum_{i,j} \left[\frac{\lambda_i}{\lambda_j} (dv_{ji})^2 + dv_{ij} dv_{ji}\right]. \tag{26}$$

Rewrite (26) as

$$d^2\mathcal{J} = \sum_{i \neq j} q_{ij} + \sum_i q_{ii}, \tag{27}$$

where

$$q_{ij} = \frac{\lambda_i}{\lambda_j} (dv_{ji})^2 + dv_{ij} dv_{ji}. \tag{28}$$

One can easily see that the summand in the second term in (27) is always positive. For a pair (i, j), $i \neq j$, the summand in the first term in (27) can be rewritten as

$$\begin{aligned} q_{ij} + q_{ji} &= \frac{\lambda_i}{\lambda_j} (dv_{ji})^2 + \frac{\lambda_j}{\lambda_i} (dv_{ij})^2 + 2dv_{ij} dv_{ji} \\ &= \begin{bmatrix} dv_{ij} & dv_{ji} \end{bmatrix} \begin{bmatrix} \frac{\lambda_j}{\lambda_i} & 1 \\ 1 & \frac{\lambda_i}{\lambda_j} \end{bmatrix} \begin{bmatrix} dv_{ij} \\ dv_{ji} \end{bmatrix} \end{aligned} \tag{29}$$

One can easily see that $q_{ij} + q_{ji}$ is always non-negative. Hence $d^2\mathcal{J}$ is always positive. Note that the stability of the algorithm (13) 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 (20).

IV. Numerical Examples

We have performed experiments with 3 digitized voice signals, all of which are sampled at 8 kHz. Three mixture signals were generated using the mixing matrix given by

$$\boldsymbol{A} = \begin{bmatrix} 0.224 & 0.055 & 0.469 \\ 0.162 & 0.505 & 0.476 \\ 0.933 & 0.649 & 0.912 \end{bmatrix}. \tag{30}$$

We evaluate the performance of three algorithms:

Algorithm 1: the Matsuoka's algorithm in [3].

Algorithm 2: the natural gradient algorithm (13) for feedback network.

Algorithm 3: the natural gradient algorithm (20) for feedforward network.

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.

As performance measure, we use the performance index (PI) defined by

$$\begin{aligned} \text{PI} &= \sum_{i=1}^{n} \left\{ \left(\sum_{k=1}^{n} \frac{|g_{ik}|}{\max_j |g_{ij}|} - 1 \right) \right. \\ &\quad \left. + \left(\sum_{k=1}^{n} \frac{|g_{ki}|}{\max_j |g_{ji}|} - 1 \right) \right\}, \end{aligned}$$

where g_{ij} is the (i,j)th element of the global system matrix $\boldsymbol{G}$ ($\boldsymbol{G} = (\boldsymbol{I} - \boldsymbol{W})^{-1}\boldsymbol{A}$ for a recurrent network, $\boldsymbol{G} = \widehat{\boldsymbol{W}}\boldsymbol{A}$ for a feedforward network) and $\max_j g_{ij}$ represents the maximum value among the elements in the ith row vector of $\boldsymbol{G}$, $\max_j g_{ji}$ does the maximum value among the elements in the ith column vector of $\boldsymbol{G}$. The performance index defined in (31) tells us how far the global system matrix $\boldsymbol{G}$ is from a generalized permutation matrix. When perfect signal separation is achieved, the performance index is zero. In practice, when the performance index falls below 0.01, the separation is satisfactory.

In addition to the performance measure (31), we also calculated the Signal to Interference Ratio Improvement ($SIRI$) defined by

$$SIRI_i = \frac{E\{(x_i - s_i)^2\}}{E\{(y_i - s_i)^2\}}. \tag{31}$$

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 (31).

Numerical experimental results are shown in Fig. 1 and are summarized in I. Poor performance of Algorithm 1 might result from the simplification approximation made in [3] which is not reasonable for the case of $n \geq 3$. Moreover, our algorithms posses the equivariant property, thus they give satisfactory results even for the case of ill-conditioned mixing. More details can be found in [16].

V. Conclusions

We have presented two natural gradient learning algorithm which perform second-order nonstationary source separation. We also presented local stability analysis of the algorithms and showed that separating solutions are always locally stable stationary points of the proposed algorithms, regardless of probability distributions of sources. Numerical experimental results confirmed the high performance of the algorithms.

Type of Algorithm	SIRI
Matsuoka [3]	$SIRI_1$= 16.0dB
	$SIRI_2$= 14.8dB
	$SIRI_3$= 36.5dB
Feedback (13)	$SIRI_1$= 68.1dB
	$SIRI_2$= 61.5dB
	$SIRI_3$= 65.1dB
Feedforward (20)	$SIRI_1$= 68.0dB
	$SIRI_2$= 61.5dB
	$SIRI_3$= 65.2dB

TABLE I
PERFORMANCE COMPARISON IN TERMS OF SIRI.

VI. Acknowledgment

This work was supported by Korea Ministry of Science and Technology under an International Cooperative Research Project and Brain Science and Engineering Research Program and by Korea Ministry of Information and Communication under Advanced backbone IT technology development project and by Ministry of Education of Korea for its financial support toward the Electrical and Computer Engineering Division at POSTECH through its BK21 program.

References

[1] S. Haykin, *Unsupervised Adaptive Filtering: Blind Source Separation*. Prentice-Hall, 2000.

[2] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. John Wiley & Sons, Inc., 2001.

[3] K. Matsuoka, M. Ohya, and M. Kawamoto, "A neural net for blind separation of nonstationary signals," *Neural Networks*, vol. 8, no. 3, pp. 411–419, 1995.

[4] S. Choi and A. Cichocki, "Blind separation of nonstationary and temporally correlated sources from noisy mixtures," in

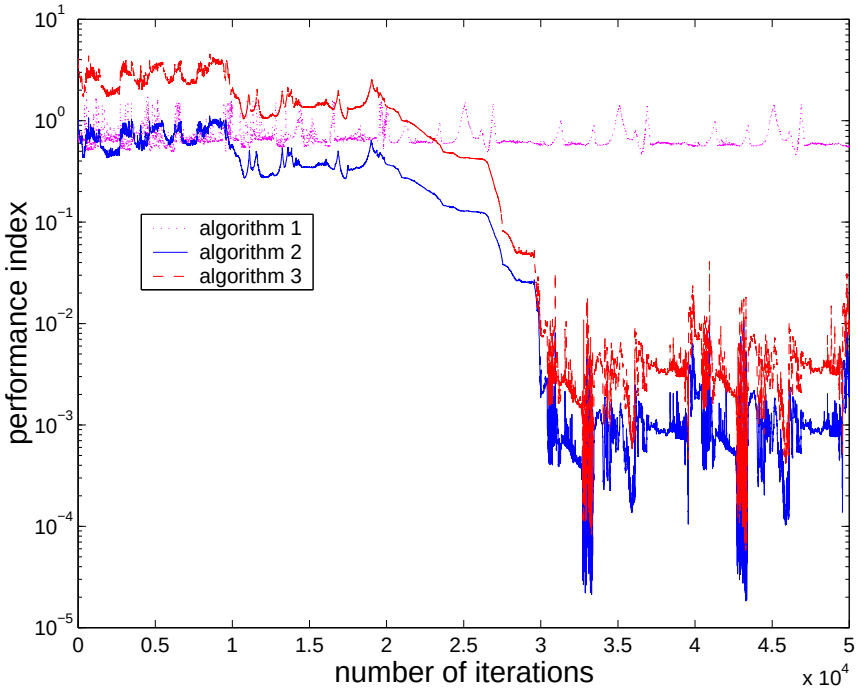

Fig. 1. The evolution of performance index is shown.

Proc. IEEE Workshop on Neural Networks for Signal Processing, (Sidney, Austrailia), pp. 405–414, 2000.

[5] C. Chang, Z. Ding, S. F. Yau, and F. H. Y. Chan, "A matrix-pencil approach to blind separation of colored nonstationary signals," *IEEE Trans. Signal Processing*, vol. 48, pp. 900–907, Mar. 2000.

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

[7] S. Choi, A. Cichocki, and A. Belouchrani, "Second order nonstationary source separation," *Journal of VLSI Signal Processing*, 2002, to appear.

[8] D. T. Pham and J. F. Cardoso, "Blind separation of instantaneous mixtures of nonstationary sources," *IEEE Trans. Signal Processing*, vol. 49, pp. 1837–1848, Sep. 2001.

[9] J. F. Cardoso and B. H. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Processing*, vol. 44, no. 12, pp. 3017–3030, Dec. 1996.

[10] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, no. 2, pp. 251–276, 1998.

[11] S. Amari, S. C. Douglas, A. Cichocki, and H. H. Yang, "Multichannel blind deconvolution and equalization using the natural gradient," in *Proc. SPAWC*, (Paris, France), pp. 101–104, 1997.

[12] S. Choi, S. Amari, and A. Cichocki, "Natural gradient learning for spatio-temporal decorrelation: Recurrent network," *IEICE Trans. Fundamentals*, vol. E83-A, pp. 2715–2722, Dec. 2000.

[13] A. Cichocki and R. Unbehauen, "Robust neural networks with on-line learning for blind identification and blind separation of sources," *IEEE Trans. Circuits and Systems - I: Fundamental Theory and Applications*, vol. 43, pp. 894–906, 1996.

[14] S. Amari, T. P. Chen, and A. Cichocki, "Nonholonomic orthogonal learning algorithms for blind source separation," *Neural Computation*, vol. 12, no. 6, pp. 1463–1484, 2000.

[15] S. Amari, T. P. Chen, and A. Cichocki, "Stability analysis of learning algorithms for blind source separation," *Neural Networks*, vol. 10, no. 8, pp. 1345–1351, 1997.

[16] S. Choi, A. Cichocki, and S. Amari, "Equivariant nonstationary source separation," *Neural Networks*, 2002, to appear.