

Flexible Independent Component Analysis

SEUNGJIN CHOI

*Department of Electrical Engineering
Chungbuk National University, KOREA
Email: schoi@engine.chungbuk.ac.kr*

ANDRZEJ CICHOCKI

*Brain-Style Information Systems Research Group
Brain Science Institute, RIKEN, JAPAN
Email: cia@brain.riken.go.jp
Warsaw University of Technology, POLAND*

SHUN-ICHI AMARI

*Brain-Style Information Systems Research Group
Brain Science Institute, RIKEN, JAPAN
Email: amari@brain.riken.go.jp*

;

Editors: M. Van Hulle, M. Niranjana, and T. Adali

Abstract. This paper addresses an independent component analysis (ICA) learning algorithm with flexible nonlinearity, so named as *flexible ICA*, that is able to separate instantaneous mixtures of sub- and super-Gaussian source signals. In the framework of natural Riemannian gradient, we employ the parameterized generalized Gaussian density model for hypothesized source distributions. The nonlinear function in the flexible ICA algorithm is controlled by the Gaussian exponent according to the estimated kurtosis of demixing filter output. Computer simulation results and performance comparison with existing methods are presented.

.1. Introduction

Independent component analysis is a statistical method that plays an important role in lots of applications such as telecommunications [13, 14], feature extraction [8, 9, 17], biomedical signal analysis [37, 32], and data analysis [27] where multiple sensors are involved. The task of ICA is to extract statistically independent components from their linear mixtures without resorting to any prior knowledge. It is known that ICA performs blind source separation under mild conditions [24, 3]. In other words, we can recover sources blindly from their linear instantaneous mixtures by a linear transformation that transforms sensor signals to the output signals that are statistically independent. The demixing system can be viewed as a recognition model in the context of machine learning.

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

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

where $\mathbf{s}(t) = [s_1(t), \dots, s_n(t)]^T$ 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 mutually independent non-Gaussian signals. The number of sensors, m is greater than or equal to the number of sources, n .

The goal of ICA is to recover source signal vector $\mathbf{s}(t)$ from the observation vector $\mathbf{x}(t)$ without the knowledge of \mathbf{A} nor $\mathbf{s}(t)$. To perform this task, we find a linear mapping \mathbf{W} which forces statistical dependence among the output signals $\{y_i(t)\}$ to be minimized. It is well known that due to the lack of prior information, there are two indeterminacies in ICA [24]: (1) scaling ambiguity; (2) permutation ambiguity. That is, the recovered signal vector $\mathbf{y}(t)$ has the form $\mathbf{y}(t) = \mathbf{P}\mathbf{\Lambda}\mathbf{s}(t)$, where \mathbf{P} is a permutation matrix and $\mathbf{\Lambda}$ is some nonsingular diagonal scaling matrix. In many applications, waveforms of sources are important factors.

Since Jutten and Herault's [33] first solution to ICA, several methods have been proposed. They include robust neural networks approach [23, 22], information maximization [7], natural gradient learning [6, 5], maximum likelihood estimation [41, 36, 40, 11], equivariant algorithms [12], nonlinear principal component analysis (PCA) [34, 39, 31], blind signal extraction [25, 21], cross-cumulants method [18, 38, 15]. Mutual information minimization, information maximization, maximum likelihood estimation result in an identical optimization function (loss function) [11].

Typical ICA learning algorithms rely on the choice of nonlinear functions, the optimal form of which depends on probability distributions of sources. Since probability distributions of sources are not known in advance in ICA task, we count on the hypothesized density models for sources. Especially for the mixtures of sub- and super-Gaussian sources, the smart choice of nonlinearity is essential. To this end, several algorithms have been developed [29, 26, 20, 28, 35]. In the present paper, we employ the generalized Gaussian density model that can approximate both super- and sub-Gaussian sources by the appropriate choice of Gaussian exponent. Preliminary result was reported in [16]. This is an extended version of our work [16] with some new results.

This paper is organized as follows. Next section is devoted to give a brief review of natural gradient based ICA algorithms. In Section .3, the generalized Gaussian density model is introduced. We also discuss the relation between the Gaussian exponent and the kurtosis in the generalized Gaussian distribution. In the framework of natural gradient based ICA algorithms, a smart way to select a nonlinear function is introduced in Section .4. Practical implementation of the flexible ICA algorithms are also presented in Section .4. Stability with several different nonlinear functions is studied in Section .5. Computer simulation results with artificial data and real world data are presented in Section .6. Conclusions with some discussions are drawn in Section .7

.2. Natural Riemannian Gradient Based ICA Algorithms

Gradient descent learning is a popular method for the purpose of minimizing a given loss function. When a parameter space (on which a loss 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 [1]. Recently the *natural gradient* was proposed by Amari [1] and was shown to be efficient in on-line learning. See [1] for more details of the natural gradient. Note that the relative gradient developed independently by Cardoso and Laheld [12] is identical to the natural gradient in the context of ICA. In this section, we briefly review two natural gradient based ICA algorithms.

.2.1. Natural Riemannian Gradient

Let us consider a linear network whose output $\mathbf{y}(t)$ is described by

$$\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t), \quad (2)$$

where (i, j) th element of the matrix \mathbf{W} , i.e., w_{ij} represents a synaptic weight between $y_i(t)$ and $x_j(t)$. In the limit of zero noise, for the square ICA problem (equal number of sources and sensors, the result can be easily extended to the case $m > n$), maximum likelihood or mutual information minimization leads to the following loss function [6, 11]:

$$L(\mathbf{W}) = -\log |\det \mathbf{W}| - \sum_{i=1}^n \log p_i(y_i), \quad (3)$$

where $p_i(\cdot)$ represent the probability density function. Let us define

$$\varphi_i(y_i) = -\frac{d \log p_i(y_i)}{dy_i}. \quad (4)$$

With this definition, the gradient of the loss function (3) is

$$\begin{aligned} \nabla L(\mathbf{W}) &= \frac{\partial L(\mathbf{W})}{\partial \mathbf{W}} \\ &= -\mathbf{W}^{-T} + \varphi(\mathbf{y})\mathbf{x}^T, \end{aligned} \quad (5)$$

where $\varphi(\mathbf{y})$ is the element-wise function whose i th component is $\varphi_i(y_i)$.

The natural Riemannian gradient (denoted by $\tilde{\nabla} L(\mathbf{W})$) learning algorithm for \mathbf{W} is given by [1, 12, 22]

$$\begin{aligned} \mathbf{W}(t+1) &= \mathbf{W}(t) - \eta_t \tilde{\nabla} L(\mathbf{W}) \\ &= \mathbf{W}(t) - \eta_t \frac{\partial L(\mathbf{W})}{\partial \mathbf{W}} \mathbf{W}^T(t) \mathbf{W}(t) \\ &= \mathbf{W}(t) + \eta_t \{\mathbf{I} - \varphi(\mathbf{y}(t))\mathbf{y}^T(t)\} \mathbf{W}(t). \end{aligned} \quad (6)$$

.2.2. Natural Riemannian Gradient in Orthogonality Constraint

Natural Riemannian gradient in orthogonality constraint has been recently proposed by Amari [2]. Let us assume that the observation vector $\mathbf{x}(t)$ has already been whitened by preprocessing and source signals are normalized, i.e.,

$$E\{\mathbf{x}(t)\mathbf{x}^T(t)\} = \mathbf{I}_m, \quad (7)$$

$$E\{\mathbf{s}(t)\mathbf{s}^T(t)\} = \mathbf{I}_n. \quad (8)$$

From (7) and (8), we have

$$\mathbf{A}\mathbf{A}^T = \mathbf{I}_m. \quad (9)$$

The m row vectors of \mathbf{A} are orthogonal n dimensional unit vectors. The set of n dimensional subspaces in \mathbb{R}^m is called Stiefel manifold. The natural Riemannian gradient in the Stiefel manifold was calculated by Amari [2]

$$\tilde{\nabla} L(\mathbf{W}) = \nabla L(\mathbf{W}) - \mathbf{W}\{\nabla L(\mathbf{W})\}^T \mathbf{W}. \quad (10)$$

Using this result, the natural gradient is given by

$$\tilde{\nabla} L(\mathbf{W}) = \varphi(\mathbf{y})\mathbf{x}^T - \mathbf{y}\varphi^T(\mathbf{y})\mathbf{W}. \quad (11)$$

Then the learning algorithm for \mathbf{W} is given by

$$\begin{aligned}\mathbf{W}(t+1) &= \mathbf{W}(t) - \eta_t \tilde{\nabla} L(\mathbf{W}) \\ &= \mathbf{W}(t) - \eta_t \{ \varphi(\mathbf{y}(t)) \mathbf{x}^T(t) - \mathbf{y}(t) \varphi^T(\mathbf{y}(t)) \mathbf{W}(t) \}.\end{aligned}\quad (12)$$

It should be noted that when $m = n$, the matrix \mathbf{W} is orthogonal in each iteration step, so this reduces to the following form

$$\mathbf{W}(t+1) = \mathbf{W}(t) - \eta_t \{ \varphi(\mathbf{y}(t)) \mathbf{y}^T(t) - \mathbf{y}(t) \varphi^T(\mathbf{y}(t)) \} \mathbf{W}(t).\quad (13)$$

In practice, due to the skew-symmetry of the term $\varphi(\mathbf{y}(t)) \mathbf{y}^T(t) - \mathbf{y}(t) \varphi^T(\mathbf{y}(t))$, decorrelation (or whitening) processing can be performed simultaneously together with separation. With taking this into account, the algorithm becomes Cardoso and Laheld's EASI algorithm [12]

$$\Delta \mathbf{W}(t) = \eta_t \{ \mathbf{I} - \mathbf{y}(t) \mathbf{y}^T(t) - \varphi(\mathbf{y}(t)) \mathbf{y}^T(t) + \mathbf{y}(t) \varphi^T(\mathbf{y}(t)) \} \mathbf{W}(t).\quad (14)$$

The algorithms aforementioned belong to a class of on-line learning algorithms which is based on stochastic approximation. We can also consider the batch versions of the algorithms by estimating time average instead of instantaneous realization. For example, the batch version of the algorithm (14) is given by

$$\Delta \mathbf{W}(t) = \eta_t \{ \mathbf{I} - \langle \mathbf{y}(t) \mathbf{y}^T(t) \rangle - \langle \varphi(\mathbf{y}(t)) \mathbf{y}^T(t) \rangle + \langle \mathbf{y}(t) \varphi^T(\mathbf{y}(t)) \rangle \} \mathbf{W}(t),\quad (15)$$

where $\langle \cdot \rangle$ denotes the time average operation.

3. Generalized Gaussian Density Model for Sources

Optimal nonlinear function $\varphi_i(y_i)$ is given by (4). However, it requires the knowledge of the probability distributions of sources which are not available to us. A variety of hypothesized density model has been used. For example, for super-Gaussian source signals, unimodal or hyperbolic-Cauchy distribution model [36] leads to the nonlinear function given by

$$\varphi_i(y_i) = \tanh(\beta y_i).\quad (16)$$

Such sigmoid function was also used in [7]. For sub-Gaussian source signals, cubic nonlinear function $\varphi_i(y_i) = y_i^3$ has been a favorite choice. For mixtures of sub- and super-Gaussian source signals, according to the estimated kurtosis of the extracted signals, nonlinear function can be selected from two different choices [26]. Several approaches [29, 20, 28, 35] are already available.

This paper present a flexible nonlinear function derived using generalized Gaussian density model [20, 19, 16]. It will be shown that the nonlinear function is self-adaptive and controlled by the Gaussian exponent.

3.1. The Generalized Gaussian Distribution

The *generalized Gaussian* probability distribution is a set of distributions parameterized by a positive real number α , which is usually referred to as the *Gaussian exponent* of the distribution. The Gaussian exponent α controls the "peakiness" of the distribution. The probability density function (PDF) for a generalized Gaussian is described by

$$p(y; \alpha) = \frac{\alpha}{2\lambda\Gamma(\frac{1}{\alpha})} e^{-|\frac{y}{\lambda}|^\alpha},\quad (17)$$

where $\Gamma(x)$ is Gamma function given by

$$\Gamma(x) = \int_0^\infty t^{x-1} e^{-t} dt.\quad (18)$$

Note that if $\alpha = 1$, the distribution becomes the standard ‘‘Laplacian’’ distribution. If $\alpha = 2$, the distribution is standard normal distribution (see Figure 1).

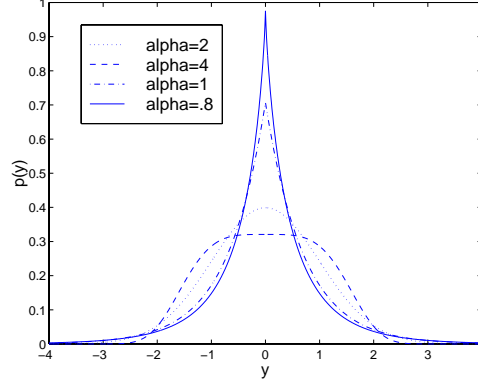


Fig. 1. The generalized Gaussian distribution is plotted for several different values of Gaussian exponent, $\alpha = 0.8, 1, 2, 4$.

3.2. The Moments of the Generalized Gaussian Distribution

In order to fully understand the generalized Gaussian distribution, it is useful to look at its moments (specially 2nd and 4th moments which give the kurtosis). The n th moment of the generalized Gaussian distribution is given by

$$M_n = \int_{-\infty}^{\infty} y^n p(y; \alpha) dy. \quad (19)$$

If n is odd, the integrand is the product of an even function and an odd function over the whole real line, which integrates to zero. In particular, this implies that the mean of the distribution given in (17) is zero and it is symmetric about its mean (which means its skewness is zero).

The even moments, on the other hand, completely characterize the distribution. In computing these moments, we use the following integral formula (see pp. 386 in [30])

$$\int_0^{\infty} y^{\nu-1} e^{-\mu y^a} dy = \frac{1}{a} \mu^{-\frac{1}{a}} \Gamma\left(\frac{\nu}{a}\right). \quad (20)$$

The 2nd moment of the generalized Gaussian distribution is determined by

$$\begin{aligned} M_2 &= \int_{-\infty}^{\infty} y^2 p(y; \alpha) dy \\ &= 2 \int_0^{\infty} y^2 \frac{\alpha}{2\lambda \Gamma\left(\frac{1}{\alpha}\right)} e^{-|\frac{y}{\lambda}|^\alpha} dy. \end{aligned} \quad (21)$$

We are integrating only over the positive values of y , we can remove the absolute value in the exponent. Thus

$$M_2 = \frac{\alpha}{\lambda \Gamma\left(\frac{1}{\alpha}\right)} \int_0^{\infty} y^2 e^{-(\frac{y}{\lambda})^\alpha} dy. \quad (22)$$

Making the substitution $z = \frac{y}{\lambda}$ ($dy = \lambda dz$), we find

$$M_2 = \frac{\alpha \lambda^2}{\Gamma\left(\frac{1}{\alpha}\right)} \int_0^{\infty} z^2 e^{-z^\alpha} dz. \quad (23)$$

Invoking the integral formula (20), we have

$$M_2 = \lambda^2 \frac{\Gamma\left(\frac{3}{\alpha}\right)}{\Gamma\left(\frac{1}{\alpha}\right)}. \quad (24)$$

In similar way, we can find the 4th moment given by

$$M_4 = \lambda^4 \frac{\Gamma\left(\frac{5}{\alpha}\right)}{\Gamma\left(\frac{1}{\alpha}\right)}. \quad (25)$$

In general, the $(2k)$ th moment is given by

$$M_{2k} = \lambda^{2k} \frac{\Gamma\left(\frac{2k+1}{\alpha}\right)}{\Gamma\left(\frac{1}{\alpha}\right)}. \quad (26)$$

3.3. Kurtosis and Gaussian Exponent

The kurtosis is a nondimensional quantity. It measures the relative peakedness or flatness of a distribution. A distribution with positive kurtosis is termed *leptokurtic* (super-Gaussian). A distribution with negative kurtosis is termed *platykurtic* (sub-Gaussian). The kurtosis of the distribution is defined in terms of the 2nd- and 4th-order moments as

$$\kappa(y) = \frac{M_4}{M_2^2} - 3, \quad (27)$$

where the constant term -3 makes the value zero for standard normal distribution.

For a generalized Gaussian distribution, the kurtosis can be expressed in terms of the Gaussian exponent, given by

$$\kappa_\alpha = \frac{\Gamma\left(\frac{5}{\alpha}\right) \Gamma\left(\frac{1}{\alpha}\right)}{\Gamma^2\left(\frac{3}{\alpha}\right)} - 3. \quad (28)$$

The plot of kurtosis κ_α versus the Gaussian exponent α for leptokurtic and platykurtic signals are shown in Figure 2.

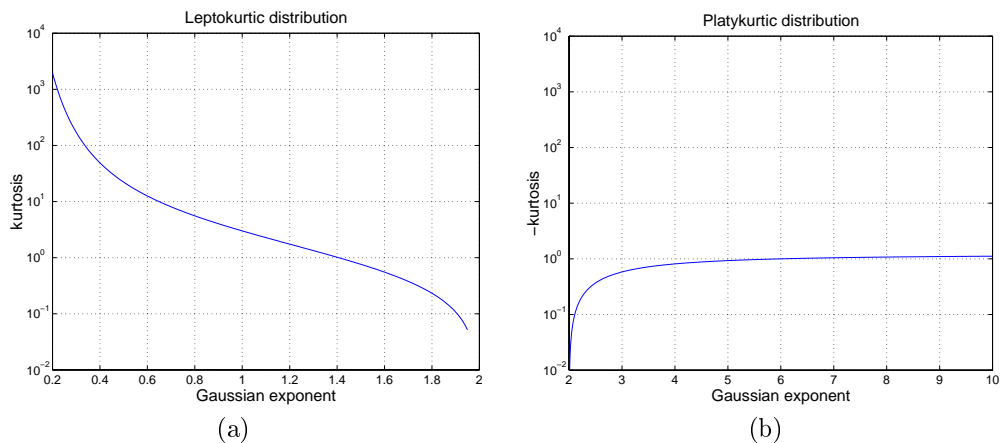


Fig. 2. The plot of kurtosis κ_α versus Gaussian exponent α : (a) for leptokurtic signal; (b) for platykurtic signal.

.4. The Flexible ICA Algorithm

From the parameterized generalized Gaussian density model, the nonlinear function in the algorithm (14) is given by

$$\begin{aligned}\varphi_i(y_i) &= \frac{d \log p_i(y_i)}{dy_i} \\ &= |y_i|^{\alpha_i-1} \text{sgn}(y_i),\end{aligned}\quad (29)$$

where $\text{sgn}(y_i)$ is the signum function of y_i .

Note that for $\alpha_i = 1$, $\varphi_i(y_i)$ in (29) becomes a signum function (which can also be derived from the Laplacian density model for sources). The signum nonlinear function is favorable for the separation of speech signals since natural speeches is often modeled as Laplacian distribution. Note also that for $\alpha_i = 4$, $\varphi_i(y_i)$ in (29) becomes a cubic function, which is known to be a good choice for sub-Gaussian sources.

In order to select a proper value of the Gaussian exponent α_i , we estimate the kurtosis of the output signal y_i and select the corresponding α_i from the relationship in Figure 2. The kurtosis of y_i , κ_i can be estimated via the following iterative algorithm:

$$\kappa_i(t+1) = \frac{M_{4i}(t+1)}{M_{2i}^2(t+1)} - 3, \quad (30)$$

where

$$M_{4i}(t+1) = (1 - \delta)M_{4i}(t) + \delta|y_i(t)|^4, \quad (31)$$

$$M_{2i}(t+1) = (1 - \delta)M_{2i}(t) + \delta|y_i(t)|^2, \quad (32)$$

where δ is a small constant, say, 0.01.

In general, the estimated kurtosis of demixing filter output does not exactly match the kurtosis of original source. However, it provides an idea whether the estimated source is sub-Gaussian signal or super-Gaussian signal. Moreover, it was shown [11, 3] that the performance of source separation is not degraded even if the hypothesized density does not match the true density. From these reasons, we suggest a practical method where only several different forms of nonlinear functions are used.

The kurtosis of platykurtic source does not change much as the Gaussian exponent varies (see Figure 2 (b)), so we use $\alpha_i = 4$ if the estimated kurtosis of y_i is negative. The cubic nonlinearity for sub-Gaussian source is also involved with the kurtosis minimization method [12]. For leptokurtic source, one can see the kurtosis varies much according to the Gaussian exponent (see Figure 2 (a)). Thus we suggest several different values of α_i , in contrast to the case of sub-Gaussian source. From our experience, two or three different values of the Gaussian exponent are enough to handle various super-Gaussian sources. Typical examples of nonlinear functions with different values of α_i are shown in Figure 3.

.5. Local Stability Analysis

The stability conditions for the algorithm (6) and the algorithm (14) were given by Amari *et al.* [4] and by Cardoso and Laheld [12], respectively. We first briefly review some important results obtained in [4, 12, 10]. Then we investigate the stability effect of several nonlinear functions that we have derived from the generalized Gaussian density model. To this end, we focus on the algorithm (14) which employs the natural gradient in Stiefel manifold.

Since the algorithm (14) was derived from the gradient $dL = \varphi^T(\mathbf{y})d\mathbf{W}\mathbf{x}$, we need to calculate its Hessian d^2L to check the stability of stationary points. Amari *et al.* has shown that the calculation of Hessian d^2L is relatively easier if the modified differential coefficient matrix $d\mathbf{Z} = d\mathbf{W}\mathbf{W}^{-1}$ is employed. Note that the modified differential coefficient matrix $d\mathbf{Z}$ is skew-symmetric in the orthogonality constraint,

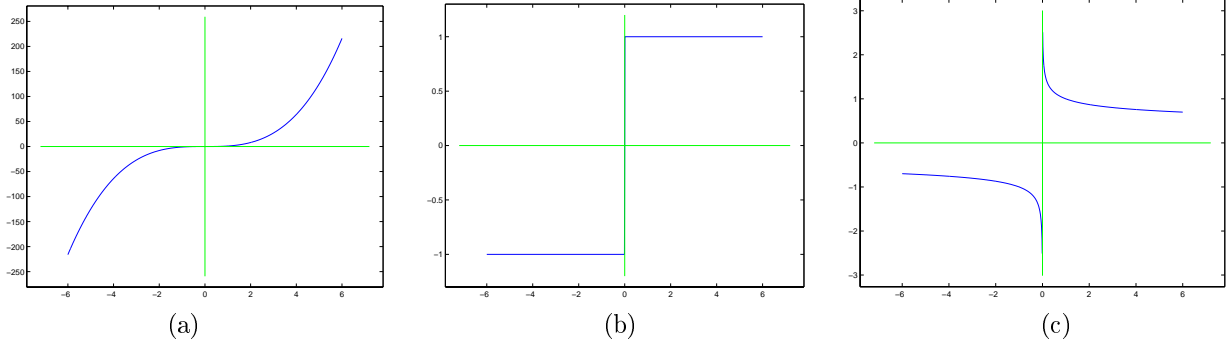


Fig. 3. The exemplary shape of nonlinear function $\varphi_i(y_i)$: (a) for $\alpha_i = 4$; (b) for $\alpha_i = 1$; (c) for $\alpha_i = 0.8$.

we calculate the Hessian d^2L . Using the fact that $d\mathbf{Z}$ is skew-symmetric, dL can be written as

$$\begin{aligned}
 dL &= \varphi^T(\mathbf{y})d\mathbf{y} \\
 &= \varphi^T(\mathbf{y})d\mathbf{Z}\mathbf{y} \\
 &= \sum_{i,j} \varphi_i(y_i)dz_{ij}y_j \\
 &= \sum_{i>j} \{\varphi_i(y_i)y_j - \varphi_j(y_j)y_i\} dz_{ij}.
 \end{aligned} \tag{33}$$

Then the Hessian d^2L is calculated as

$$\begin{aligned}
 d^2L &= \sum_{i>j} \sum_k \{\dot{\varphi}_i(y_i)dz_{ik}y_ky_j - \dot{\varphi}_j(y_j)dz_{jk}y_ky_i \\
 &\quad + \varphi_i(y_i)dz_{jk}y_k - \varphi_j(y_j)dz_{ik}y_k\} dz_{ij}.
 \end{aligned} \tag{34}$$

Taking into account the normalization constraint ($E\{y_i^2\} = E\{y_j^2\} = 1$) and the skew-symmetry, $dz_{ij} = -dz_{ji}$, the expected Hessian at $\mathbf{W} = \mathbf{A}^{-1}$ (at desirable solution) is given by

$$\begin{aligned}
 E\{d^2L\} &= \sum_{i>j} [E\{\dot{\varphi}_i(y_i)\} + E\{\dot{\varphi}_j(y_j)\} \\
 &\quad - E\{\varphi_i(y_i)y_i\} - E\{\varphi_j(y_j)y_j\}] dz_{ij}^2,
 \end{aligned} \tag{35}$$

where $\dot{\varphi}_i(y_i)$ denotes the derivative of $\varphi_i(y_i)$ with respect to y_i . From (34), the stability condition is given by

$$\chi_i + \chi_j > 0, \tag{36}$$

where

$$\chi_i = E\{\dot{\varphi}_i(y_i)\} - E\{\varphi_i(y_i)y_i\}. \tag{37}$$

The stability condition given in (36) coincides with that in [10] but we arrived at this result in the framework of the natural gradient in Stiefel manifold. For each y_i , the condition

$$\chi_i > 0 \tag{38}$$

is a sufficient condition for stability.

Several different nonlinear functions were suggested in the flexible ICA algorithm. Here we investigate the stability of stationary points of the algorithm (14) for three different cases: (1) $\alpha_i = 4$ for $\kappa_i < 0$; (2) $\alpha_i = 1$; (3) $\alpha_i = .8$ for $\kappa_i > 0$.

.5.1. *Case 1: $\alpha_i = 4$*

The choice of $\alpha_i = 4$ was suggested for sub-Gaussian source ($\kappa_i < 0$). The choice of $\alpha_i = 4$ results in the cubic nonlinear function, i.e., $\varphi_i(y_i) = |y_i|^2 y_i$. With this selection, one can easily see that the lefthand side of (38) is the kurtosis of y_i multiplied by -1. Since y_i is sub-Gaussian, the condition (38) is satisfied.

.5.2. *Case 2: $\alpha_i = 1$*

With the choice of $\alpha_i = 1$, the generalized Gaussian density (17) becomes Laplacian density, i.e.,

$$p_i(y_i) = \frac{1}{2\lambda_i} e^{-|y_i/\lambda_i|}. \quad (39)$$

The choice of $\alpha_i = 1$ leads to the signum function (hard limiter), i.e.,

$$\begin{aligned} \varphi_i(y_i) &= \text{sgn}(y_i) \\ &= \frac{y_i}{|y_i|}. \end{aligned} \quad (40)$$

In order to calculate the derivative of the signum function, we model it as the sum of two unit step functions, i.e.,

$$\text{sgn}(y_i) = u(y_i) - u(-y_i), \quad (41)$$

where $u(y_i)$ is the unit step function. Then we can calculate the derivative, $\dot{\varphi}_i(y_i)$

$$\dot{\varphi}_i(y_i) = 2\delta(y_i). \quad (42)$$

We compute $E\{\dot{\varphi}_i(y_i)\}$

$$\begin{aligned} E\{\dot{\varphi}_i(y_i)\} &= \int_{-\infty}^{\infty} 2\delta(y_i) \frac{1}{2\lambda_i} e^{-|y_i/\lambda_i|} dy_i \\ &= \frac{1}{\lambda_i}. \end{aligned} \quad (43)$$

We also compute $E\{\varphi_i(y_i)y_i\}$

$$\begin{aligned} E\{\varphi_i(y_i)y_i\} &= E\{|y_i|\} \\ &= \lambda_i. \end{aligned} \quad (44)$$

The normalized constraint, $E\{y_i^2\} = 1$ gives

$$E\{y_i^2\} = 2\lambda_i^2 = 1. \quad (45)$$

Then, we have $\lambda_i = \sqrt{\frac{1}{2}}$. Note that χ_i is given by

$$\chi_i = \frac{1 - \lambda_i^2}{\lambda_i}. \quad (46)$$

Since $\lambda_i = \sqrt{\frac{1}{2}}$, χ_i is positive for $\kappa_i > 0$.

.5.3. *Case 3: $\alpha_i < 1$*

For highly peaky sources ($\kappa_i \gg 1$), it might be desirable to choose the value of α_i less than 1. This gives a non-increasing nonlinear function. With this choice, the nonlinear function is singular around the origin.

Thus in practical application, for $y_i \in [-\epsilon, \epsilon]$ where ϵ is very small positive number, the corresponding nonlinear function is restricted to have constant values.

The variance of y_i for the generalized Gaussian distribution is given by

$$E\{y_i^2\} = \lambda_i^2 \frac{\Gamma\left(\frac{3}{\alpha_i}\right)}{\Gamma\left(\frac{1}{\alpha_i}\right)}. \quad (47)$$

From the normalization constraint, $E\{y_i^2\} = 1$, λ_i has the following value,

$$\lambda_i = \sqrt{\frac{\Gamma\left(\frac{1}{\alpha_i}\right)}{\Gamma\left(\frac{3}{\alpha_i}\right)}}. \quad (48)$$

Besides the region for $y_i \in [-\epsilon, \epsilon]$, we can compute $E\{\dot{\varphi}_i(y_i)\}$ and $E\{\varphi_i(y_i)y_i\}$ given by

$$\begin{aligned} E\{\dot{\varphi}_i(y_i)\} &= \int_{-\infty}^{\infty} (\alpha_i - 2)|y_i|^{(\alpha_i-2)} \frac{\alpha_i}{2\lambda_i\Gamma\left(\frac{1}{\alpha_i}\right)} e^{-\frac{|y_i|^{\alpha_i}}{\lambda_i^{\alpha_i}}} dy_i \\ &= \frac{(\alpha_i - 2)\lambda_i^{\alpha_i-2}}{\Gamma\left(\frac{1}{\alpha_i}\right)} \Gamma\left(\frac{\alpha_i - 1}{\alpha_i}\right), \\ E\{\varphi_i(y_i)y_i\} &= \int_{-\infty}^{\infty} y_i|y_i|^{(\alpha_i-1)} \text{sgn}(y_i) \frac{\alpha_i}{2\lambda_i\Gamma\left(\frac{1}{\alpha_i}\right)} e^{-\frac{|y_i|^{\alpha_i}}{\lambda_i^{\alpha_i}}} dy_i \\ &= \frac{\alpha_i\lambda_i^{\alpha_i+1}}{\Gamma\left(\frac{1}{\alpha_i}\right)} \frac{1}{\alpha_i} \Gamma\left(\frac{\alpha_i + 1}{\alpha_i}\right). \end{aligned} \quad (49)$$

Note that the gamma function $\Gamma(x)$ has many singular points especially for $x < 0$. Thus special care is required with the choice of $\alpha_i < 1$. For instance, the choice of $\alpha_i = 0.5$ does not satisfy the condition (38) since $\Gamma(-1) = \infty$. For the case of $\alpha_i = 0.8$, one can easily see that the stability condition (38) is satisfied.

.6. Experimental Results

.6.1. Artificial Data

We have performed an experiment with two super-Gaussian sources and two sub-Gaussian sources (see Figure 4). The kurtoses of sources are -1.2, -1.5, 3.3, 3.6. They were artificially mixed using the mixing matrix \mathbf{A} given by

$$\mathbf{A} = \begin{bmatrix} 0.155 & 0.204 & 0.431 & 0.739 \\ 0.526 & 0.511 & 0.404 & 0.614 \\ 0.205 & 0.392 & 0.306 & 0.941 \\ 0.141 & 0.937 & 0.656 & 0.182 \end{bmatrix}. \quad (50)$$

The Gaussian exponent α can be learned from the estimated kurtosis of the demixing filter output, through the relation as shown in 2. However, the estimated kurtosis does not exactly match the true one, but its sign does. In practice, only several different values of α can be employed in learning process. In this experiment, three different values of Gaussian exponent α were used: (1) $\alpha = .8$ when the estimated kurtosis of recovered signal $y_i(t)$ is greater than 20; (2) $\alpha = 1$ when the estimated kurtosis of recovered signal is between 0 and 20; (3) $\alpha = 4$ when the estimated kurtosis of recovered signal is negative.

As a performance measure, we have used the performance index defined by

$$PI = \sum_{i=1}^n \left\{ \left(\sum_{k=1}^n \frac{|g_{ik}|^2}{\max_j |g_{ij}|^2} - 1 \right) + \left(\sum_{k=1}^n \frac{|g_{ki}|^2}{\max_j |g_{ji}|^2} - 1 \right) \right\}, \quad (51)$$

where g_{ij} is the (i, j) th element of the global system matrix $\mathbf{G} = \mathbf{W}\mathbf{A}$ and $\max_j g_{ij}$ represents the maximum value among the elements in the i th row vector of \mathbf{G} , $\max_j g_{ji}$ does the maximum value among the elements in the i th column vector of \mathbf{G} . When perfect signal separation is carried out, the performance index PI is zero. In practice, it is very small number.

The synaptic weight matrix \mathbf{W} was initialized as the identity matrix. The learning rate $\eta_t = 0.0005$ was used. Performance comparison was made with extended infomax ICA algorithm [35, 28] where the nonlinear function is switched between fixed $-\tanh(\cdot)$ and $\tanh(\cdot)$ according to the sign of estimated kurtosis.

Mixtures and recovered signals by the flexible ICA algorithm (14) and the extended infomax are shown in Figures 5, 6, 7. Performance comparison between the flexible ICA algorithm and the extended infomax algorithm is shown in Figure 8. In Figure 8, the batch versions of both algorithms were used and prewhitening of data was not performed for both algorithms for fair comparison. One can observe that the flexible ICA algorithm gives faster convergence and better performance. Faster convergence might be due to the natural gradient in Stiefel manifold, i.e., the decorrelation is performed together with separation. Better performance might result from the flexible nonlinear function controlled by the Gaussian exponent in the flexible ICA algorithm in contrast to the fixed nonlinear function employed by the extended infomax.

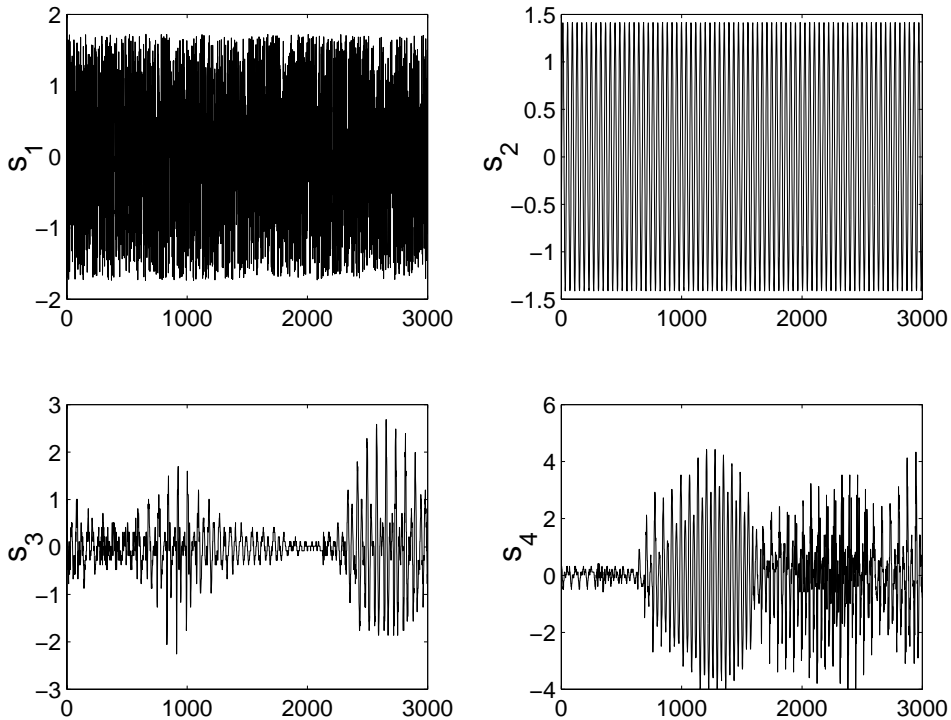


Fig. 4. Original source signals.

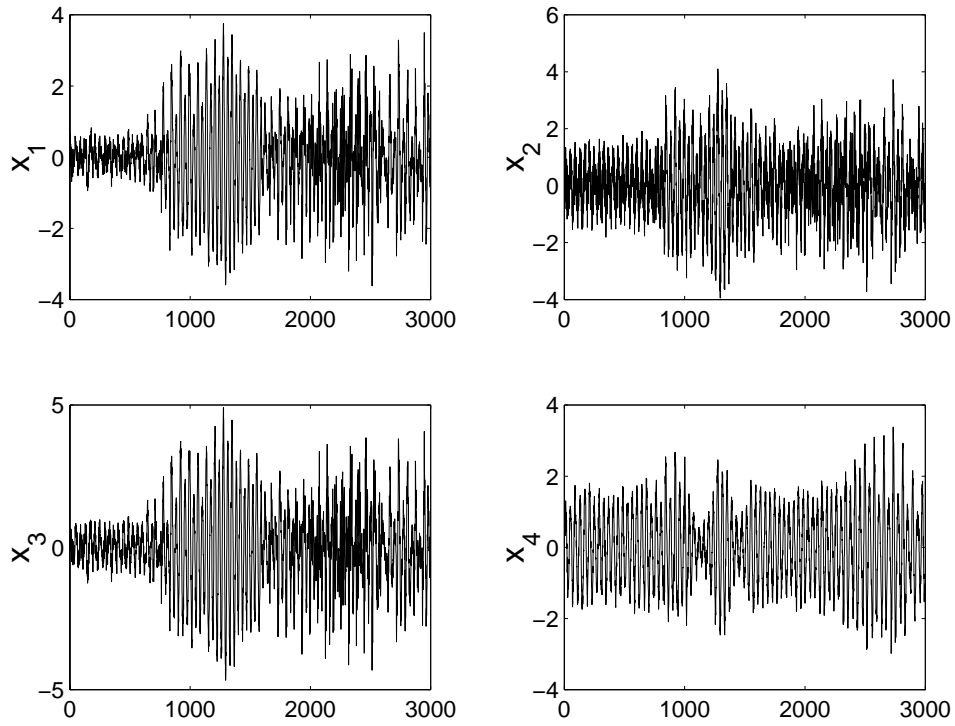


Fig. 5. Mixture signals.

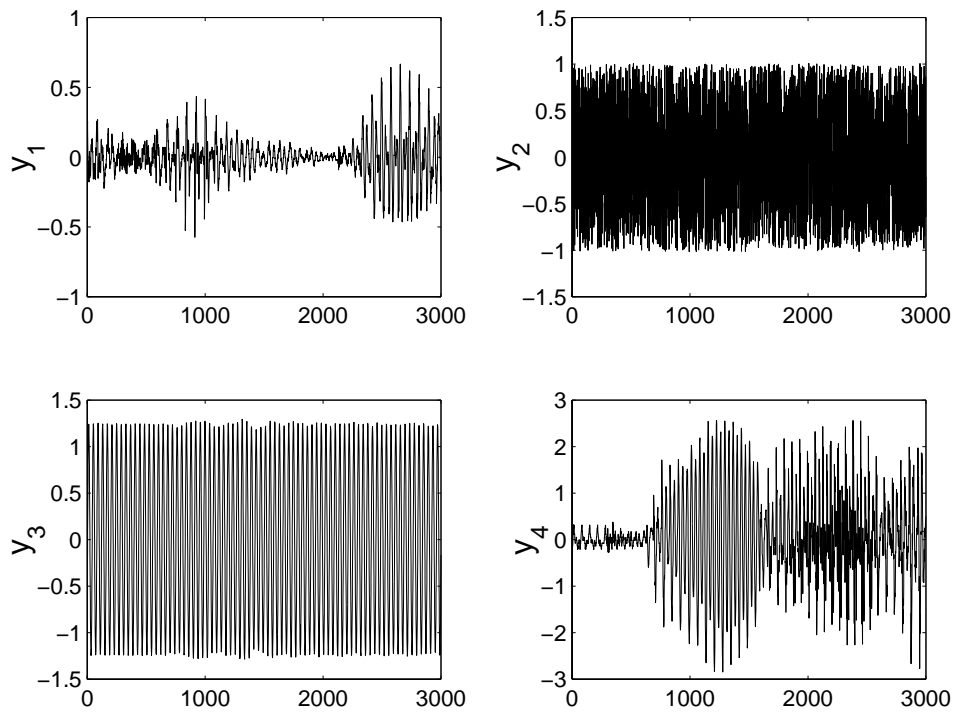


Fig. 6. Recovered signals by the flexible ICA algorithm.

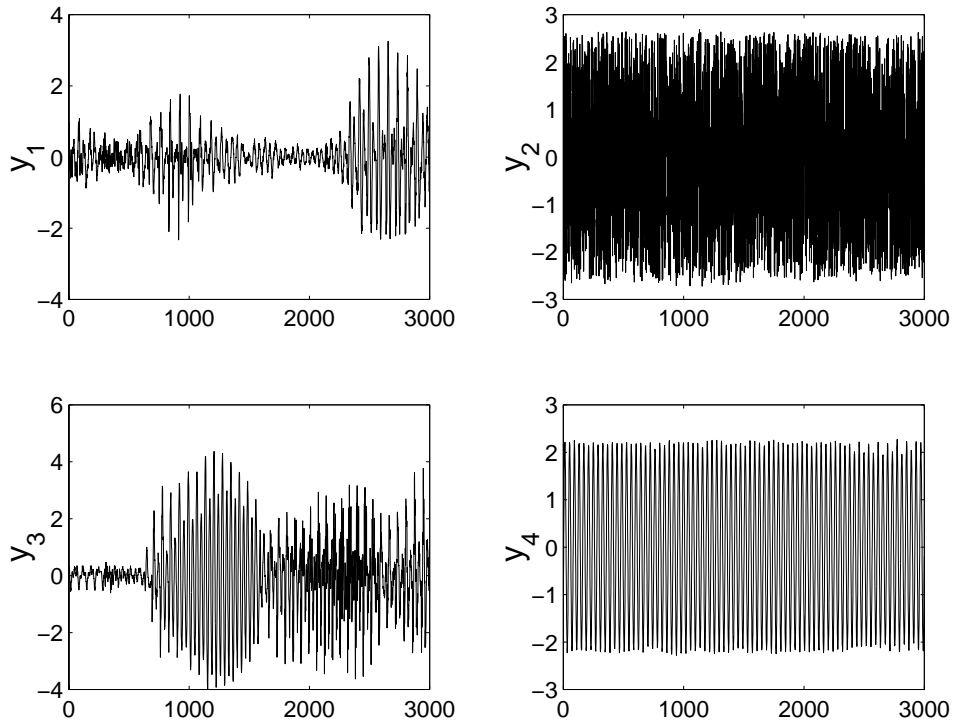


Fig. 7. Recovered signals by the extended infomax algorithm.

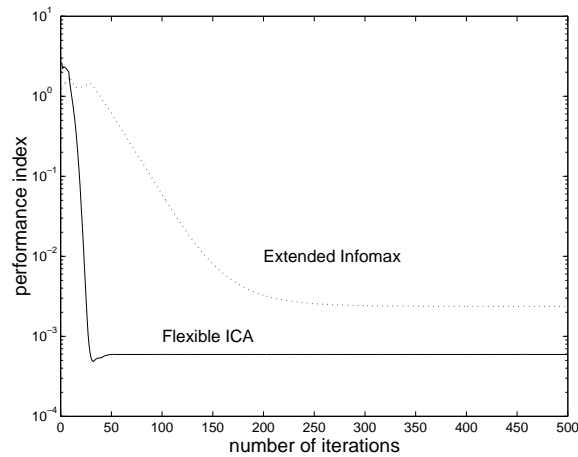


Fig. 8. The evolution of performance index: solid line is for the flexible ICA algorithm and dotted line is for the extended infomax algorithm.

.6.2. Extraction of Fetal ECG source

The ECG data as shown in Figure 9 are the potential recordings during an 8-channel experiment. Only 5 seconds of recordings (resampled at 500 Hz) are displayed. In this experiment, the electrodes were placed on the abdomen and the cervix of the mother. Abdominal signals measured near fetus are shown in channel 1 to 5. The weak fetal contributions are contained in x_1 to x_5 , although they are not clearly visible. The ECG raw data measured through 8 channels are dominated by mother's ECG (MECG).

In order to enhance or separate FECG, principal component analysis (PCA) was applied and the result are shown in Figure 9. The PCA aims at finding a orthogonal transformation which best models the covariance structure of the data. First two principal components are MECG, and the third principal component might be FECG, but it is not clear. Since only second-order statistics is used in PCA, it is not possible to separate MECG and FECG from raw data.

The flexible ICA algorithm was applied to process the ECG raw data, and the result in shown in Figure 9. The 3rd node output signal y_3 corresponds to the FECG signal. Breathing artifact is well extracted at the 4th output node, y_4 . The 2nd and 8th node contain the MECG. The rest of extracted signals might contain noise contributions. The weak FECG signal was well extracted by the flexible ICA algorithm, whereas the PCA had a difficulty to extract it. We also applied the extended infomax algorithm to this data set and the result is shown in Figure 9. It can be observed that the extracted FECG signal was not as clear as the one by the flexible ICA algorithm.

.7. Conclusions

We have presented the flexible ICA algorithm (in the framework of the natural Riemannian gradient) where self-adaptive nonlinear function is used. For the hypothesized density model, we have employed the generalized Gaussian distribution that is able to model most uni-modal probability distribution. The nonlinear function in the algorithm was derived from the generalized Gaussian density. In contrast most existing methods, the nonlinear function in our algorithm is controlled by a single parameter (Gaussian exponent). As a practical and simple method, we have suggested several different nonlinear functions that resulted from different values of the Gaussian exponent and confirmed the validity of our approach through computer simulations. In addition, rigorous stability analysis for several nonlinear functions was carried out.

.8. Acknowledgment

This work was supported in part by the Braintech 21, Ministry of Science and Technology, KOREA and in part by BSI, RIKEN, JAPAN.

References

1. S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, Feb. 1998.
2. S. Amari. Natural gradient for over- and under-complete bases in ICA. *Neural Computation*, 11(8):1875–1883, Nov. 1999.
3. S. Amari and J. F. Cardoso. Blind source separation: Semiparametric statistical approach. *IEEE Trans. Signal Processing*, 45:2692–2700, 1997.
4. S. Amari, T. P. Chen, and A. Cichocki. Stability analysis of learning algorithms for blind source separation. *Neural Networks*, 10(8):1345–1351, 1997.
5. S. Amari and A. Cichocki. Adaptive blind signal processing - neural network approaches. *Proc. of IEEE, Special Issue on Blind Identification and Estimation*, 86(10):2026–2048, October 1998.
6. S. Amari, A. Cichocki, and H. H. Yang. A new learning algorithm for blind signal separation. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 757–763. MIT press, 1996.

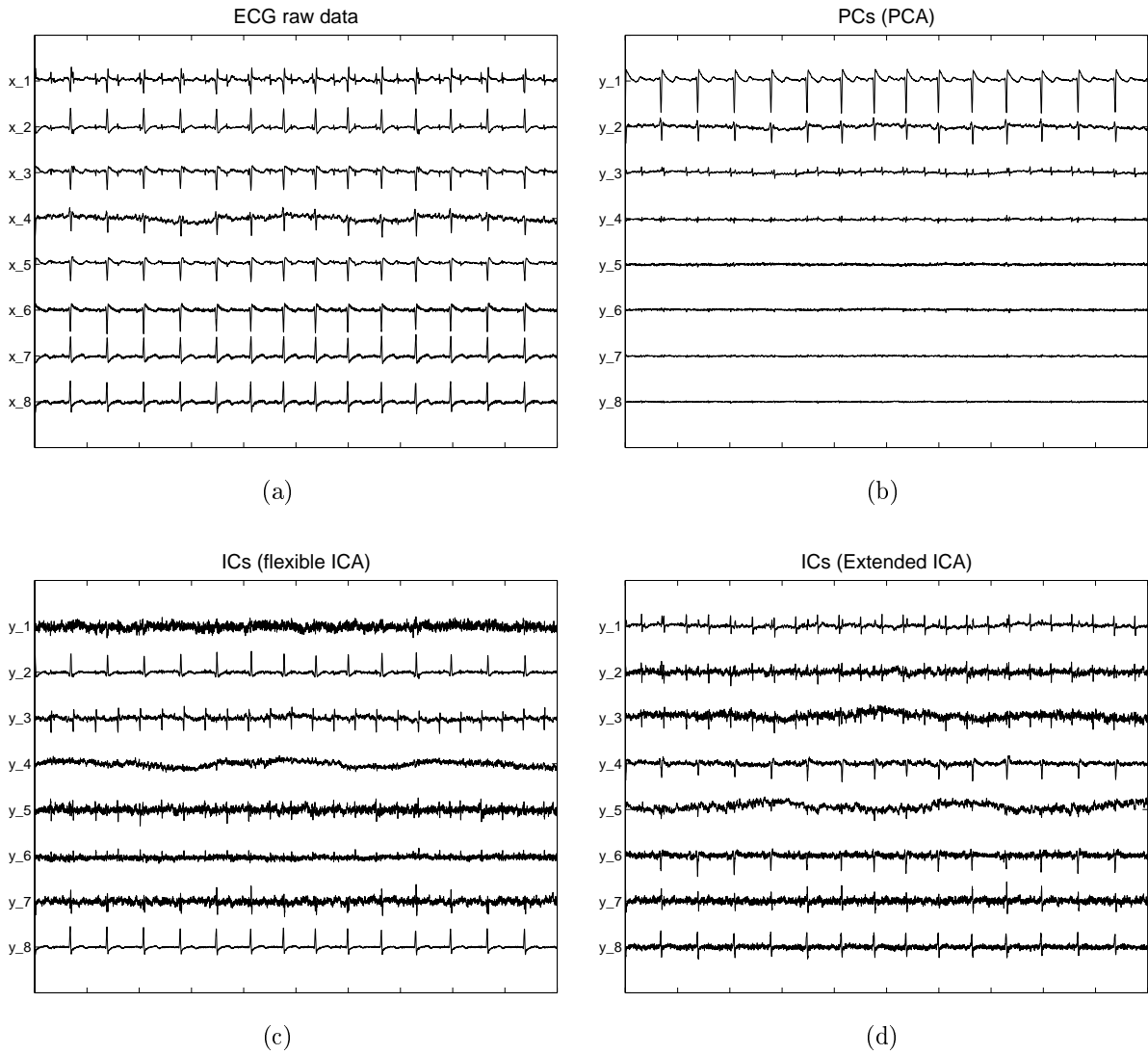


Fig. 9. Experimental result with ECG data: (a) ECG raw data; (b) principal components in a descending order from top to bottom; (c) independent components extracted by the flexible ICA algorithm; (d) independent components extracted by the extended infomax algorithm.

7. A. Bell and T. Sejnowski. An information maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7:1129–1159, 1995.
8. A. Bell and T. Sejnowski. Learning the higher-order structure of a natural sound. *Network: Computation in Neural Systems*, 7:261–266, 1996.
9. A. Bell and T. Sejnowski. The independent components of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997.
10. J. -F. Cardoso. On the stability of source separation algorithms. In T. Constantinides, S. Y. Kung, M. Niranjan, and E. Wilson, editors, *Neural Networks for Signal Processing VIII*, pages 13–22, 1998.
11. J. -F. Cardoso. Infomax and maximum likelihood for source separation. *IEEE Signal Processing Letters*, 4(4):112–114, Apr. 1997.
12. J. -F. Cardoso and B. H. Laheld. Equivariant adaptive source separation. *IEEE Trans. Signal Processing*, 44(12):3017–3030, Dec. 1996.
13. J. -F. Cardoso and A. Souloumiac. Blind beamforming for non Gaussian signals. *IEE Proceedings-F*, 140(6):362–370, 1993.
14. S. Choi. Adaptive blind signal separation for multiuser communications: An information-theoretic approach. *Journal of Electrical Engineering and Information Science*, 4(2):249–256, April 1999.
15. S. Choi and A. Cichocki. A linear feedforward neural network with lateral feedback connections for blind source separation. In *IEEE Signal Processing Workshop on Higher-order Statistics*, pages 349–353, Banff, Canada, 1997.
16. S. Choi, A. Cichocki, and S. Amari. Flexible independent component analysis. In T. Constantinides, S. Y. Kung, M. Niranjan, and E. Wilson, editors, *Neural Networks for Signal Processing VIII*, pages 83–92, 1998.
17. S. Choi, A. Cichocki, and S. Amari. Fetal electrocardiogram data analysis using flexible independent component analysis. In *The 4th Asia-Pacific Conference on Medical and Biological Engineering (APCMBE'99)*, Seoul, Korea, 1999.
18. S. Choi, R. Liu, and A. Cichocki. A spurious equilibria-free learning algorithm for the blind separation of non-zero skewness signals. *Neural Processing Letters*, 7:61–68, 1998.
19. A. Cichocki, I. Sabala, and S. Amari. Intelligent neural networks for blind signal separation with unknown number of sources. In *Int. Symp. Engineering of Intelligent Systems*, pages 148–154, Tenerife, Spain, 1998.
20. A. Cichocki, I. Sabala, S. Choi, B. Orsier, and R. Szupiluk. Self-adaptive independent component analysis for sub-Gaussian and super-Gaussian mixtures with unknown number of source signals. In *International Symposium on Nonlinear Theory and Applications*, pages 731–734, 1997.
21. A. Cichocki, R. Thawonmas, and S. Amari. Sequential blind signal extraction in order specified by stochastic properties. *Electronics Letters*, 33(1):64–65, 1997.
22. 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*, 43:894–906, 1996.
23. A. Cichocki, R. Unbehauen, and E. Rummert. Robust learning algorithm for blind separation of signals. *Electronics Letters*, 43(17):1386–1387, 1994.
24. P. Comon. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, 1994.
25. N. Delfosse and P. Loubaton. Adaptive blind separation of independent sources: A deflation approach. *Signal Processing*, 45:59–83, 1995.
26. S. C. Douglas, A. Cichocki, and S. Amari. Multichannel blind separation and deconvolution of sources with arbitrary distributions. In J. Principe, L. Giles, N. Morgan, and E. Wilson, editors, *Neural Networks for Signal Processing VII*, pages 436–445, 1997.
27. M. Girolami. Hierarchic dichotomizing of polychotomous data - an ICA based data mining tool. In *First International Workshop on Independent Component Analysis and Signal Separation*, pages 197–201, 1999.
28. M. Girolami. An alternative perspective on adaptive independent component analysis algorithms. *Neural Computation*, 10(8):2103–2114, Nov. 1998.
29. M. Girolami and C. Fyfe. Generalized independent component analysis through unsupervised learning with emergent Busgang properties. In *Proc. ICNN*, pages 1788–1791, 1997.
30. I. S. Gradshteyn, I. M. Ryzhik, and A. Jeffrey. *Table of Integrals, Series, and Products*. Academic Press, 1994.
31. A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9:1483–1492, 1997.
32. T. P. Jung, C. Humphries, T. Lee, S. Makeig, M. McKeown, V. Iragui, and T. Sejnowski. Extended ICA removes artifacts from electroencephalographic recordings. In *Advances in Neural Information Processing Systems*, volume 10, pages 894–900, 1998.
33. C. Jutten and J. Herault. Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10, 1991.
34. J. Karhunen. Neural approaches to independent component analysis. In *European Symposium on Artificial Neural Networks*, pages 249–266, 1996.
35. T. W. Lee, M. Girolami, and T. Sejnowski. Independent component analysis using an extended infomax algorithm for mixed sub-Gaussian and super-Gaussian sources. *Neural Computation*, 11(2):609–633, 1999.
36. D. J. C. MacKay. Maximum likelihood and covariant algorithms for independent component analysis. Technical Report Draft 3.7, University of Cambridge, Cavendish Laboratory, 1996.

37. S. Makeig, A. Bell, T. P. Jung, and T. Sejnowski. Blind separation of auditory event-related brain responses into independent components. *Proc. of National Academy of Sciences*, 94:10979–10984, 1997.
38. J. P. Nadal and N. Parga. Redundancy reduction and independent component analysis: Conditions on cumulants and adaptive approaches. *Neural Computation*, 9:1421–1456, 1997.
39. E. Oja. The nonlinear PCA learning rule and signal separation - mathematical analysis. Technical Report A26, Helsinki University of Technology, Laboratory of Computer and Information Science, 1995.
40. B. Pearlmutter and L. Parra. Maximum likelihood blind source separation: A context-sensitive generalization of ICA. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, pages 613–619, 1997.
41. D. T. Pham. Blind separation of instantaneous mixtures of sources via an independent component analysis. *IEEE Trans. Signal Processing*, 44(11):2768–2779, 1996.

Seungjin CHOI was born in Seoul, Korea, on October 26, 1964. He received the B.S. and M.S. degrees in electrical engineering from Seoul National University, Korea, in 1987 and 1989, respectively and the Ph.D degree in electrical engineering from the University of Notre Dame, Indiana, in 1996.

After spending the fall of 1996 as a Visiting Assistant Professor in the Department of Electrical Engineering at University of Notre Dame, Indiana, he was as Frontier Researcher with the Laboratory for Artificial Brain Systems, RIEKN in Japan. In August 1997, he joined the School of Electrical and Electronics Engineering at Chungbuk National University where he is currently an Assistant Professor. He has also been an Invited Senior Research Fellow at Laboratory for Open Information Systems, Brain-style Information Systems Research Group in Brain Science Institute, RIKEN in Japan. His current research interests include brain information processing, statistical (blind) signal processing, independent component analysis, unsupervised learning, and multiuser communications.

Andrzej CICHOCKI was born in Poland on August 1947. He received the M.Sc.(with honors), Ph.D., and Habilitate Doctorate (Dr.Sc.) degrees, all in electrical engineering and computer science, from Warsaw University of Technology (Poland) in 1972, 1975, and 1982, respectively.

Since 1972, he has been with the Institute of Theory of Electrical Engineering and Electrical Measurements at the Warsaw University of Technology, where he became a full Professor in 1995.

He is the co-author of two international books: *MOS Switched-Capacitor and Continuous-Time Integrated Circuits and Systems* (Springer-Verlag, 1989) and *Neural Networks for Optimization and Signal Processing* (J Wiley and Teubner Verlag, 1993/94) and author or co-author of more than hundred fifty (150) scientific papers.

He spent at University Erlangen-Nuernberg (GERMANY) a few years as Alexander Humboldt Research Fellow and Guest Professor. In 1995-96 he has been working as a Team Leader of the Laboratory for Artificial Brain Systems, at the Frontier Research Program RIKEN (JAPAN), in the Brain Information Processing Group directed by professor Shun-ichi Amari. Currently he is head of the laboratory for Open Information Systems in the Brain Science Institute, Riken, Wako-schi, JAPAN.

He is reviewer of several international Journals, e.g. *IEEE Trans. on Neural Networks*, *Signal Processing*, *Circuits and Systems*, *Biological Cybernetics*, *Electronics Letters*, *Neurocomputing*, *Neural Computation*. He is also member of several international Scientific Committees and the associated Editor of *IEEE Transaction on Neural Networks* (since January 1998). His current research interests include signal and image processing (especially blind signal/image processing), neural networks and their electronic implementations, learning theory and algorithms, independent and principal component analysis, optimization problems, circuits and systems theory and their applications, artificial intelligence.

Shun-ichi AMARI (Fellow, IEEE) was born in Tokyo, Japan, on January 3, 1936. He graduated from the University of Tokyo in 1958, having majored in mathematical engineering, and he received the Dr.Eng. degree from the University of Tokyo in 1963.

He was an Associate Professor at Kyushu University, an Associate and then Full Professor at the Department of Mathematical Engineering and Information Physics, University of Tokyo, and is now Professor-Emeritus at the University of Tokyo. He is the Director of the Brain-Style Information Systems Group, RIKEN Brain Science Institute, Saitama Japan. He has been engaged in research in wide areas of mathematical engineering and applied mathematics, such as topological network theory, differential geometry of continuum mechanics, pattern recognition, mathematical foundations of neural networks, and information geometry.

Dr. Amari served as President of the International Neural Network Society, Council member of Bernoulli Society for Mathematical Statistics and Probability Theory, and Vice President of the Institute of Electrical, Information and Communication Engineers. He was founding Coeditor-in-Chief of Neural Networks. He has been awarded the Japan Academy Award, the IEEE Neural Networks Pioneer Award, and the IEEE Emanuel R. Piore Award.