Flexible Nonlinear Learning for Source Separation

Seungjin Choi

Abstract - Source separation is a statistical method, the goal of which is to separate the linear instantaneous mixtures of statistically independent sources without resorting to any prior knowledge. This paper addresses a source separation algorithm which is able to separate the mixtures of sub- and super-Gaussian sources. The nonlinear function in the proposed algorithm is derived from the generalized Gaussian distribution that is a set of distributions parameterized by a real positive number (Gaussian exponent). Based on the relationship between the kurtosis and the Gaussian exponent, we present a simple and efficient way of selecting proper nonlinear functions for source separation. Useful behavior of the proposed method is demonstrated by computer simulations.

Keywords - Independent component analysis, generalized Gaussian distribution, natural gradient, source separation, unsupervised learning.

1. Introduction

Source separation is a statistical method, the goal of which is to recover unknown sources (latent variables) from their linear instantaneous mixtures, without resorting to any prior information except for the assumption of statistical independence of sources. Source separation is a fundamental problem encountered in a variety of applications such as blind beamforming [1], multiuser communications [2], speech processing [3], and biomedical signal analysis [4] where multiple sensors are involved.

In the context of source separation, it is assumed that an m-dimensional observation vector $\boldsymbol{x}(t) = [\, x_1(t) \ \cdots \ x_m(t)]^T$ is generated by

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

where $A \in R^{m \times n}$ $(m \geq n)$ is called the *mixing matrix* and $\boldsymbol{s}(t)$ is an n-dimensional vector whose elements are called *sources*. The task of source separation is to design a demixing filter $\boldsymbol{W}$ such that the filter output vector $\boldsymbol{y}(t) = \boldsymbol{W}x(t)$ is a possibly re-scaled and re-ordered source vector [5], [3]. In other words, source separation seeks a linear mapping $\boldsymbol{W}$ from the observation vector $\boldsymbol{x}(t)$ to $\boldsymbol{y}(t)$ such that the transformed vector $\boldsymbol{y}(t)$ satisfies the following decomposition:

$$\begin{aligned} \boldsymbol{y}(t) &= \boldsymbol{W}x(t) \\ &= \boldsymbol{P}\Lambda\boldsymbol{s}(t) \end{aligned} \qquad (2)$$

where $\boldsymbol{P}$ is some permutation matrix and Λ is some nonsingular diagonal matrix. It is known [5] that the decomposition (2) is achieved if $\{y_i(t)\}$ are statistically independent. Along this line, source separation is closely related to the method known as *independent component analysis* (ICA), the task of which is to decompose the observation vector into a linear sum of independent components [6].

Since Jutten and Herault's first solution to source separation and ICA, several methods have been developed. These include robust neural networks [7], [8], algebraic methods [9], [5], nonlinear information maximization [3], mutual information minimization [10], nonlinear principal component analysis [11], [12], maximum likelihood estimation [13], [14], the equivariant method [15], and cross-cumulants based methods [16], [17].

In typical adaptive source separation algorithms, it is essential to choose proper nonlinear functions for successful separation. It was shown in [18] that the optimal nonlinear function is the score function that depends on the probability distribution of source. In the task of source separation, neither sources nor their probability distributions are available in advance. Thus true probability density of source is replaced by the hypothesized density model, then an appropriate nonlinear function is selected. For the separation of mixtures of super- and sub-Gaussian sources (hybrid mixtures), the smart choice of nonlinear function is necessary. Several source separation algorithms have been developed to deal with hybrid mixtures [19], [20], [21], [22]. [23]. In these methods [19], [20], [21], [22], [23], two different nonlinear functions were assumed and its selection was done according to the estimated kurtosis of demixing filter output. Two different nonlinear functions were rather heuristically adopted, or were derived from specific super- and sub-Gaussian distributions.

In the present paper, we present a general but efficient way of choosing the nonlinear function in source separation algorithms. We employ the generalized Gaussian distribution that is a set of distributions parameterized by one real positive number (Gaussian exponent). Since the generalized Gaussian distribution is able to model most uni-modal distributions, the nonlinear function is flexible in the sense that its shape varies depending on the Gaussian exponent.

Manuscript received August 30, 1999. accepted May 16, 2000.

Seungjin Choi is with the Department of Electrical Engineering, Chungbuk National University, 48 Kaeshin-dong, Cheongju, Chungbuk 361-763, Korea.

Based on the relationship between the kurtosis and the Gaussian exponent, we present a simple but efficient way of choosing the Gaussian exponent according to the estimated kurtosis.

The rest of this paper is organized as follows. Next section is devoted to give a brief review of natural (or relative) gradient source separation algorithms. In Section III, the generalized Gaussian distribution is introduced and the relationship between the kurtosis and the Gaussian exponent is discussed. In the framework of natural gradient source separation algorithms, a smart way to select a nonlinear function is introduced in Section IV. A practical method of source separation with flexible nonlinearity is also discussed in Section IV. In Section V, computer simulations are provided to demonstrate a successful result of separation of 8 sources with arbitrary distributions. A comparison with the extended Infomax algorithm [23], [24] is also made. Conclusions are drawn in Section VI.

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 [24]. Recently the *natural gradient* was proposed by Amari [24] and was shown to be efficient in on-line learning. See [24] for more details of the natural gradient. Note that the relative gradient developed independently by Cardoso and Laheld [15] is identical to the natural gradient in the context of ICA. In this section, we briefly review two natural gradient based ICA algorithms.

A. *Natural Riemannian Gradient*

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

$$\boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t) \tag{3}$$

where (i,j)th element of the matrix $\boldsymbol{W}$, 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 [10], [25]:

$$L(\boldsymbol{W}) = -\log|\det \boldsymbol{W}| - \sum_{i=1}^{n} \log p_i(y_i) \tag{4}$$

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

$$\varphi_i(y_i) = -\frac{d\log p_i(y_i)}{dy_i} \tag{5}$$

With this definition, the gradient of the loss function (4) is

$$\begin{aligned}\nabla L(\boldsymbol{W}) &= \frac{\partial L(\boldsymbol{W})}{\partial \boldsymbol{W}}\\ &= -\ \boldsymbol{W}^{-T} + \varphi(\boldsymbol{y})\ \boldsymbol{x}^T\end{aligned} \tag{6}$$

where $\varphi(y)$ is the element-wise function whose ith component is $\varphi_i(y_i)$. For shorthand notation, the time index t was dropped.

The natural Riemannian gradient (denoted by $\tilde{\nabla} L(\boldsymbol{W})$) learning algorithm for $\boldsymbol{W}$ is given by [24], [15], [8]

$$\begin{aligned}\Delta \boldsymbol{W} &= -\eta \tilde{\nabla} L(\boldsymbol{W})\\ &= -\eta \frac{\partial L(\boldsymbol{W})}{\partial \boldsymbol{W}}\ \boldsymbol{W}^T \boldsymbol{W} = \eta\{\boldsymbol{I} - \varphi(\boldsymbol{y})\ \boldsymbol{y}^T\}\boldsymbol{W}\end{aligned} \tag{7}$$

where $\eta > 0$ is a learning rate.

B. *Natural Riemannian Gradient in Orthogonality Constraint*

Natural Riemannian gradient in orthogonality constraint has been recently proposed by Amari [26]. Let us assume that the observation vector $\boldsymbol{x}$ has already been whitened by preprocessing and source signals are normalized, i.e.,

$$E\{\boldsymbol{x}\ \boldsymbol{x}^T\} = \boldsymbol{I}_m, \tag{8}$$

$$E\{s\ s^T\} = \boldsymbol{I}_n \tag{9}$$

From (8) and (9), we have

$$\boldsymbol{A}\boldsymbol{A}^T = \boldsymbol{I}_m \tag{10}$$

The m row vectors of A are orthogonal n dimensional unit vectors. The set of n dimensional subspaces in R^m is called Stiefel manifold. The natural Riemannian gradient in the Stiefel manifold was calculated by Amari [26]

$$\tilde{\nabla} L(\boldsymbol{W}) = \nabla L(\boldsymbol{W}) - \boldsymbol{W}\{\nabla L(\boldsymbol{W}))\}^T \boldsymbol{W} \tag{11}$$

Using this result, the natural gradient is given by

$$\tilde{\nabla} L(\boldsymbol{W}) = \varphi(\boldsymbol{y})\ \boldsymbol{x}^T - \boldsymbol{y}\varphi^T(\boldsymbol{y})\boldsymbol{W} \tag{12}$$

Then the learning algorithm for **W** is given by

$$\Delta \boldsymbol{W} = -\eta \tilde{\nabla} L(\boldsymbol{W})$$

$$= - \eta\{\varphi(y)\, x^T - y\varphi^T(y)\, W\} \qquad (13)$$

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

$$\Delta W = - \eta\{\varphi(y)\, y^T - y\varphi^T(y)\}\, W \qquad (14)$$

In practice, due to the skew-symmetry of the term $\varphi(y)\, y^T - y\varphi^T(y)$, 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 [15]

$$\Delta W = \eta\{I - yy^T - \varphi(y)\, y^T + y\varphi^T(y)\}\, W \qquad (15)$$

The algorithms aforementioned belong to a class of online 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 (15) is given by

$$\Delta W = \eta\{I - \langle yy^T\rangle - \langle \varphi(y)\, y^T\rangle + \langle y\varphi^T(y)\rangle\}\, W, \qquad (16)$$

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

3. Generalized Gaussian Density Model For Sources

This section introduces the generalized Gaussian distribution and reveals the relationship between the kurtosis and the Gaussian exponent.

A. 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

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

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

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

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

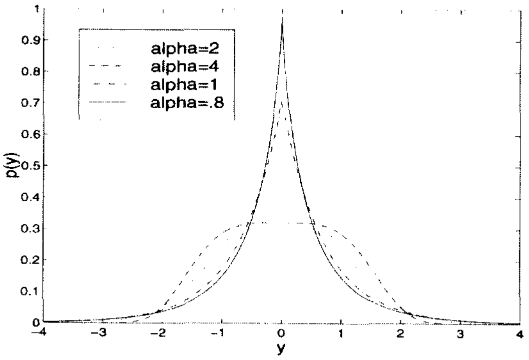


Fig. 1 The generalized Gaussian distribution is plotted for several different values of Gaussian exponent, α = 0.8, 1, 2, 4.

B. 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 nth moment of the generalized Gaussian distribution is given by

$$M_n = \int_{-\infty}^{\infty} y^n p(y, \alpha) dy \qquad (19)$$

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 [27])

$$\int_0^{\infty} y^{\nu-1} e^{-\mu y^{\alpha}} dy = \frac{1}{\alpha} \mu^{-\frac{\nu}{\alpha}} \Gamma(\frac{\nu}{\alpha}) \qquad (20)$$

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

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

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

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

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

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

Invoking the integral formula (20), we have

$$M_2 = \lambda^2 \frac{\Gamma(\frac{3}{\alpha})}{\Gamma(\frac{1}{\alpha})} \tag{24}$$

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

$$M_4 = \lambda^4 \frac{\Gamma(\frac{5}{\alpha})}{\Gamma(\frac{1}{\alpha})} \tag{25}$$

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

$$M_{2k} = \lambda^{2k} \frac{\Gamma(\frac{2k+1}{\alpha})}{\Gamma(\frac{1}{\alpha})} \tag{26}$$

C. 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

$$x(y) = \frac{M_4}{M_2^2} - 3 \tag{27}$$

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

$$x_\alpha = \frac{\Gamma(\frac{5}{\alpha})\Gamma(\frac{1}{\alpha})}{\Gamma^2(\frac{3}{\alpha})} - 3 \tag{28}$$

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

4. The Algorithm

In adaptive source separation algorithms (7) and (15), the nonlinear function $\varphi_i(y_i)$ is given by Eq. (5). The nonlinear function depends on probability distribution of source, $p_i(\cdot)$ which is not available in advance. Popular hypothesized density models are: (1) hyperbolic-Cauchy distribution (for super-Gaussian source) [28]; (2) Laplacian distribution (for super-Gaussian source) [29]; (3) Pearson mixture model (for sub-Gaussian) [22], etc. The hyperbolic-Cauchy distribution leads to the hyperbolic tangent nonlinear function (one of widely used sigmoid nonlinearity) and the Laplacian distribution gives the signum function. These nonlinear functions were known to be proper choices for super-Gaussian sources. For sub-Gaussian sources, the cubic nonlinear function has been a favorite choice. It is closely related to the minimization of kurtosis [15], [2].

For the separation of sub- and super-Gaussian mixtures, Cichocki *et al.* [21] suggested the nonlinearity-switching method where either cubic function or hyperbolic tangent function is selected, according to the sign of kurtosis of estimated source. Similarly, Girolami and Fyfe [19] suggested

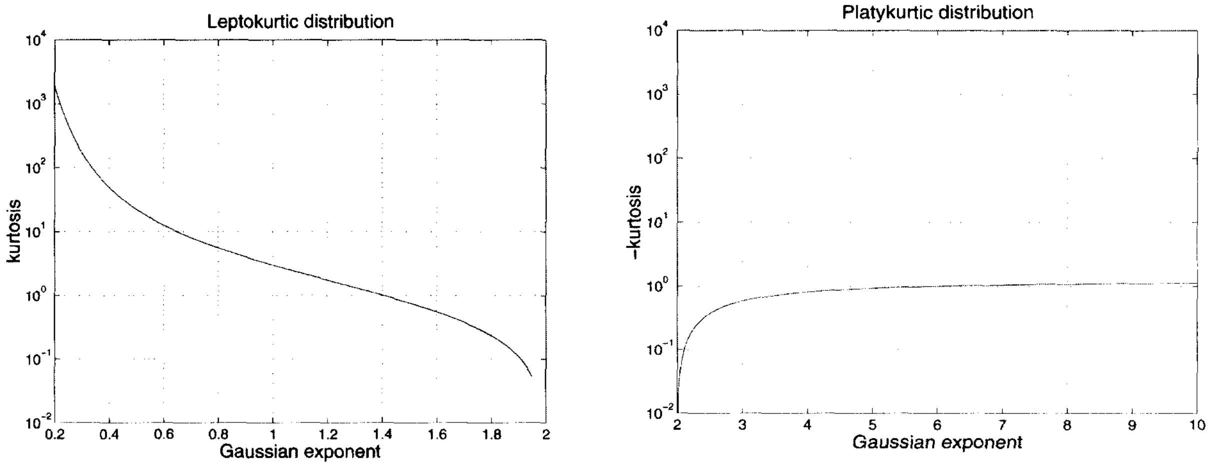


Fig. 2 The plot of kurtosis κ_α versus Gaussian exponent α: (a) for leptokurtic signal; (b) for platykurtic signal.

$\varphi(y_i) = y_i + \tanh(y_i)$ for super-Gaussian source and $\varphi(y_i) = y_i - \tanh(y_i)$ for sub-Gaussian source. This method was elaborated later in [22], [23] and is known as the extended infomax algorithm.

Now we consider the generalized Gaussian density model that was introduced in previous section. Each hypothesized density function $p_i(\cdot)$ is modeled as the generalized Gaussian density, i.e.,

$$p_i(y_i) = \frac{\alpha_i}{2\lambda\Gamma(\frac{1}{\alpha_i})} e^{-|\frac{y_i}{\lambda}|^{\alpha_i}} \qquad (29)$$

From this density model, the (normalized) nonlinear function $\varphi_i(y_i)$ is given by

$$\begin{aligned}\varphi_i(y_i) &= \frac{d \log p_i(y_i)}{dy_i} \\ &= |y_i|^{\alpha_i - 1} \operatorname{sgn}(y_i) \qquad (30)\end{aligned}$$

where $\operatorname{sgn}(y_i)$ is the signum function of y_i. Note that for $\alpha_i = 1$, (30) becomes a signum function (which can also be derived from the Laplacian density model), for $\alpha_i = 4$, (30) becomes a cubic function which is known to be a good choice for sub-Gaussian source. As described in Section III, the nonlinear function in (30) is somewhat universal and includes some popular nonlinear functions as its special cases.

The next problem is how we choose a proper value of the Gaussian exponent $\{\alpha_i\}$ for successful separation. The relationship between the kurtosis and the Gaussian exponent is summarized in Fig. 2. Once we obtain the kurtosis of estimated source, we choose a corresponding value of the Gaussian exponent according to the relation in Fig. 2. This can be done in a look-up table manner. The kurtosis of y_i, χ_i can be estimated via the following iterative algorithm:

$$\chi_i(t+1) = \frac{M_{4i}(t+1)}{M_{2i}^2(t+1)} - 3 \qquad (31)$$

where

$$M_{4i}(t+1) = (1-\delta)M_{4i}(t) + \delta|y_i(t)|^4 \qquad (32)$$

$$M_{2i}(t+1) = (1-\delta)M_{2i}(t) + \delta|y_i(t)|^2 \qquad (33)$$

where δ is a small constant, say, 0.01.

In general, the estimated kurtosis of y_i that is calculated in (31) does not exactly match the kurtosis of source s_i. However, the estimated kurtosis of y_i gives us an idea what type of signal is. In other words, at least we can see whether the demixing filter output belongs to sub-Gaussian signal, or super-Gaussian signal. It was shown in [25], [18] that the reasonable mismatch between the hypothesized density and the true density does not degrade the performance of the source separation algorithms (7) and (15). Thus it might be sufficient to use only several values of the Gaussian exponent for successful separation. For sub-Gaussian source we observe that the kurtosis does not change much with respect to the Gaussian exponent (see Fig. 2). This observation suggests only one proper choice of the Gaussian exponent may be enough for the separation of sub-Gaussian source. In practice, we choose $\alpha_i = 4$ which gives the cubic nonlinear function if the estimated kurtosis of y_i is negative. For super-Gaussian signals, one can see that kurtosis varies much with respect to the Gaussian exponent (see Fig. 2), unlike sub-Gaussian signals. Distinctively different shapes of the nonlinear function in (30) for super-Gaussian signal ($\alpha_i < 2$) occurs: (1) for $\alpha_i < 1$; (2) for $\alpha_i = 1$; (3) for $1 < \alpha_i < 2$. For example, we choose $\alpha_i = .8$, $\alpha_i = 1$, and $\alpha_i = 1.3$ (see Fig. 3). One exemplary illustration is as follows:

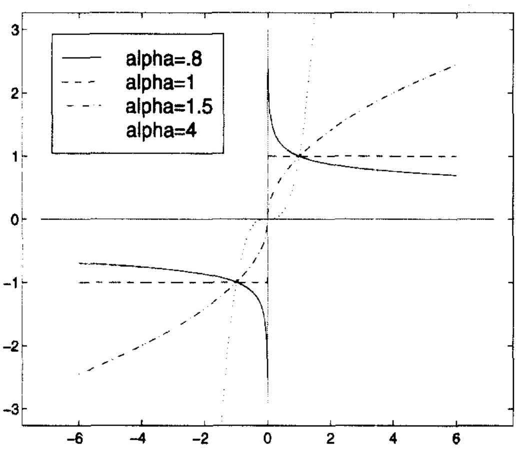


Fig. 3 The shape of nonlinear function $\varphi_i(y_i)\$$ for different values of Gaussian exponent $\alpha = 4$, 1.5, 1, 0.8

- Choose $\alpha_i = 0.8$ for $\chi_i \geq 20$.
- Choose $\alpha_i = 1$ for $5 \leq \chi_i < 20$.
- Choose $\alpha_i = 1.3$ for $0 < \chi_i < 5$.
- Choose $\alpha_i = 4$ for $\chi_i < 0$.

Here we treat the signal whose kurtosis is greater than 20, as a very spiky signal. Note that this strategy is one possible illustration. Slightly different choices are also possible, but we found that these choices worked successfully through extensive computer simulations.

5. Simulations

A. Simulation 1

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

$$\boldsymbol{A}=\begin{bmatrix}0.155 & 0.204 & 0.431 & 0.739\\ 0.526 & 0.511 & 0.404 & 0.614\\ 0.205 & 0.392 & 0.306 & 0.941\\ 0.141 & 0.937 & 0.656 & 0.182\end{bmatrix} \tag{34}$$

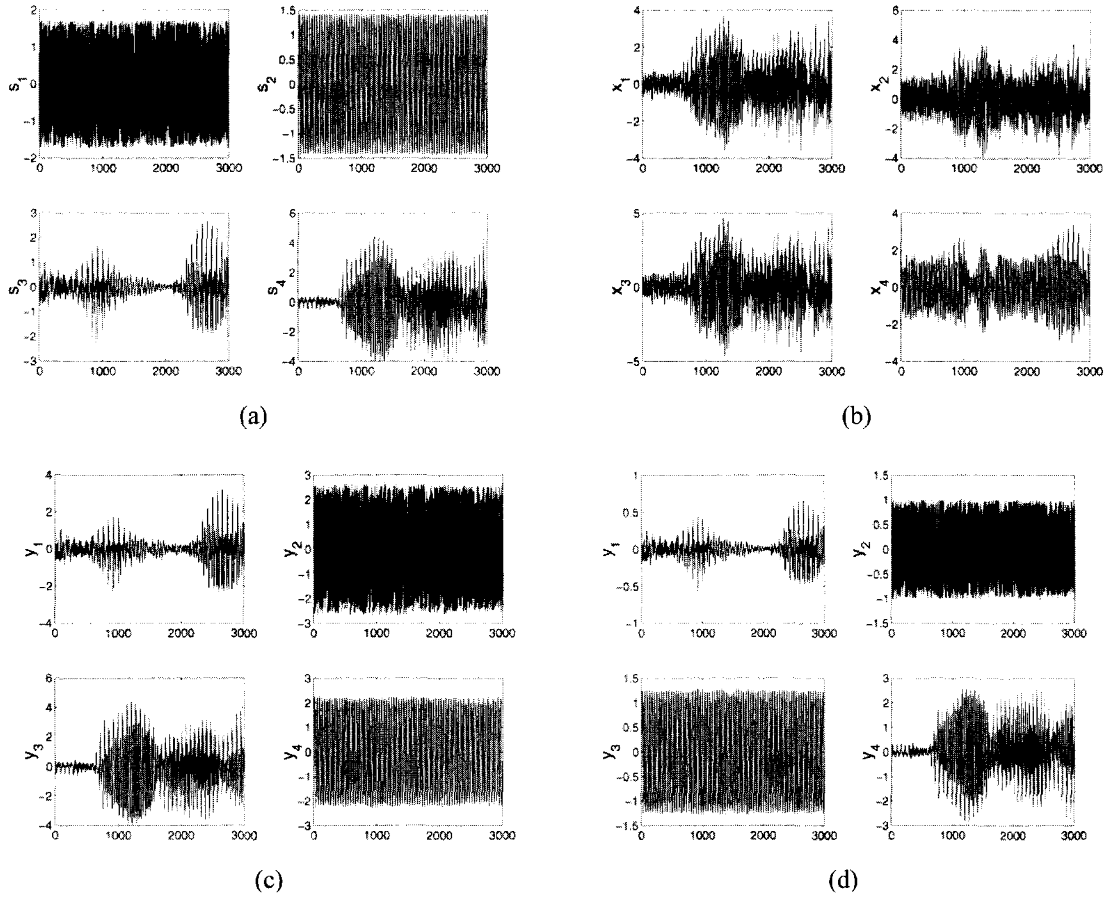
We have applied the algorithm (15) with the method of selecting Gaussian exponents $\{\alpha_i\}$ that was described in Section IV (this approach is called the flexible source separation algorithm).

As a performance measure, we have used the performance index [15], [30] defined by

$$\mathrm{PI}=\sum_{i=1}^{n}\left\{\left(\sum_{k=1}^{n}\frac{|g_{ik}|^2}{\max_j|g_{ij}|^2}-1\right)+\left(\sum_{k=1}^{n}\frac{|g_{ki}|^2}{\max_j|g_{ji}|^2}-1\right)\right\}, \tag{35}$$

where g_{ij} is the (i,j)th element of the global system matrix $\boldsymbol{G}=\boldsymbol{WA}$ 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 $\boldsymbol{G}$. When perfect signal separation is achieved, the performance index PI is zero. In practice, it is very small number.

For fair comparison with the extended infomax algorithm [23], [22], the batch versions of both algorithms were used. The learning rate $\eta=0.1$ was used and the synaptic weight matrix $\boldsymbol{W}$ was initialized as identity matrix. Mixtures and recovered signals by the flexible source separation algorithm and the extended infomax are shown in Fig. 4. Performance comparison between the flexible source separation algorithm and the extended infomax algorithm is shown in Fig. 5. In Fig. 5, 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 our flexible source separation 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 source separation algorithm in contrast to the fixed suitable nonlinear function employed by the extended infomax.



(a) (b) (c) (d)

Fig. 4 Experimental result in Simulation 1: (a) original source signals; (b) mixture signals; (c) recovered signals by the extended infomax algorithm; (d) recovered signals by the proposed algorithm (flexible ICA).

Fig. 5 The evolution of performance index in Simulation 1 : solid line is for the proposed algorithm and dotted line is for the extended infomax algorithm.

B. Simulation 2

We have performed simulations with eight different sources (see Fig. 6). Sources consist of digitized speech, the sound of ringing a bell, laughing, music, chirping, a steam-whistle, and two noises (one has uniform distribution and the other has Gaussian distribution). Sources have different probability distributions. It was assumed that sources nor their probability distributions were completely unknown in advance.

All the elements of the mixing matrix **A** were generated randomly. Then, source signals were linearly transformed by the mixing matrix. Mixture signals and their corresponding spectrograms are shown in Fig. 7.

In this simulation, we have applied the algorithm (15) with our method of selecting nonlinear functions. The learning rate $\eta = .001$ was used and the demixing filter **W** was initially set as the identity matrix. The recovered signals and their corresponding spectrograms are shown in Fig. 8. The Hinton's diagram of the elements of the global system matrix **G** is shown in Fig. 9. One can see that the matrix **G** is very closed to the generalized permutation matrix which means sources are successfully recovered.

6. Conclusions

We have presented an efficient method of learning the shapes of nonlinear functions which is essential technique for the separation of the mixtures of super- and sub-Gaussian sources. We have introduced the generalized Gaussian density model as a hypothesized density. Since the generalized Gaussian distribution is able to model most uni-modal distributions, our nonlinear function is adequate for the separation of hybrid mixtures. In contrast to most existing methods [20], [21], [19], [22], [23] where the nonlinear function is switched between two different forms according to the sign of the estimated kurtosis, the nonlinear function in our proposed method has a variety of different forms, and its shape is controlled by the Gaussian exponents whose proper value is selected from the estimated kurtosis of the demixing filter output. Useful behavior of the proposed approach was demonstrated by computer simulation results.

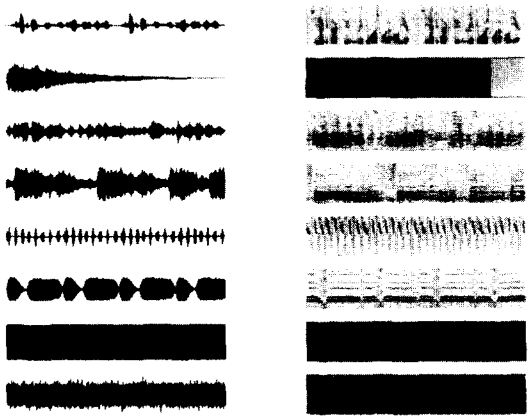


Fig. 6 Original sound sources and their corresponding spectrograms in Simulation 2.

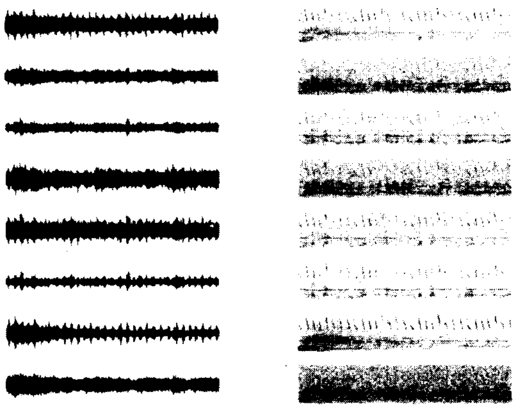


Fig. 7 Mixtures and their corresponding spectrograms in Simulation 2.

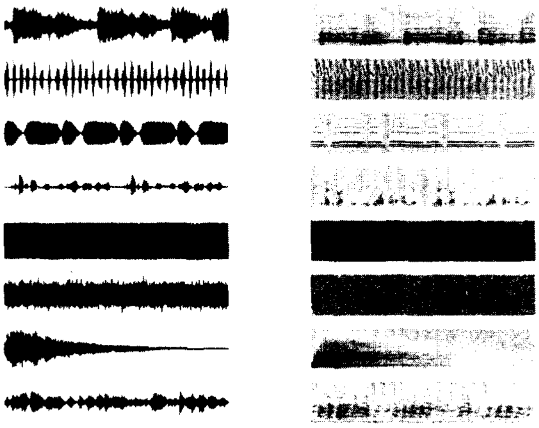


Fig. 8 Recovered signals using the proposed algorithm and their corresponding spectrograms in Simulation 2.

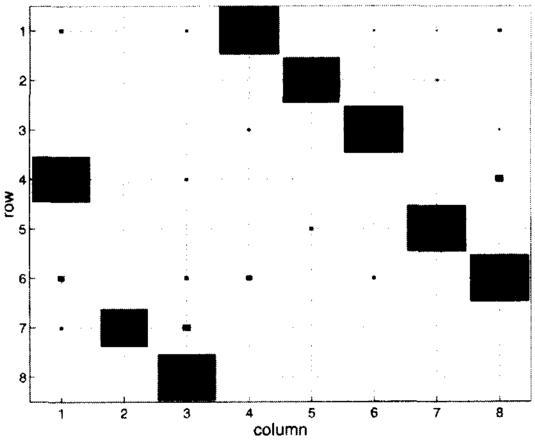


Fig. 9 Hinton diagram of the global system matrix G in Simulation 2. Each square's area represents an element's magnitude.

Acknowledgment

This work was supported by Korean Ministry of Science and Technology under Brain Science and Engineering Research Program.

References

[1] J. -F. Cardoso and A. Souloumaic, "Blind beamforming for non-Gaussian signals," *IEE Proceedings-F*, vol. 140, no. 6, pp. 362~370, 1993.

[2] S. Choi, "Adaptive blind signal separation for multiuser communications: An information-theoretic approach," *Journal of Electrical Engineering and Information Science*, vol. 4, no. 2, pp. 249~256, Apr. 1999.

[3] A. Bell and T. Sejnowski, "An information maximisation approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, pp. 1129~1159, 1995.

[4] 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*, vol. 94, pp.10979~10984, 1997.

[5] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, no. 3, pp. 287~314, 1994.

[6] S. Choi, "Neural learning algorithms for independent component analysis," *Journal of IEEE Korea Council*, vol. 2, no. 1, pp. 24~33, 1998.

[7] A. Cichocki, R. Unbehauen, and E. Rummert, "Robust learning algorithm for blind separation of signals," *Electronics Letters*, vol. 43, no. 17, pp. 1386~1387, 1994.

[8] 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.

[9] J. -F. Cardoso, "Source separation using higher-order moments," in *Proc. ICASSP*, 1989.

[10] S. Amari, A. Cichocki, and H. H. Yang, "A new learning algorithm for blind signal separation," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds. 1996, vol. 8, pp. 757~763, MIT press.

[11] E. Oja, "The nonlinear PCA learning rule and signal separation-mathematical analysis," Tech. Rep. A26, Helsinki University of Technology, Laboratory of Computer and Information Science, 1995.

[12] J. Karhunen, E. Oja, L. Wang, R. Vigario, and J. Joutsensalo, "A class of neural networks for independent component analysis," *IEEE Trans. Neural Networks*, vol. 8, pp. 486~504, 1997.

[13] D. T. Pham, P. Garrat, and C. Jutten, "Separation of mixture of independent sources through a maximum likelihood approach," in *Proc. EUSIPCO*, 1992, pp. 771~774.

[14] B. Pearlmutter and L. Parra, "Maximum likelihood blind source separation: A context-sensitive generalization of ICA," in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds., 1997, vol. 9, pp. 613~619.

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

[16] S. Choi, R. Liu, and A. Cichocki, "A spurious equlibria-free learning algorithm for the blind separatin of non-zero skewness signals," *Neural Processing Letters*, vol. 7, pp. 61~68, 1998.

[17] S. Choi and A. Cichocki, "A linear feedforward neural network with lateral feedback connections for blind source separation," in *Proc. IEEE Workshop on Higher-order Statistics*, Banff, Canada, 1997, pp. 349~353.

[18] S. Amari and J. F. Cardoso, "Blind source separation: Semi-parametric statistical approach," *IEEE Trans. Signal Processing*, vol. 45, pp. 2692~2700, 1997.

[19] M. Girolami and C. Fyfe, "Generalized independent component analysis through unsupervised learning with emergent Bussgang properties," in *Proc. ICNN*, 1997, pp. 1788~1791.

[20] S. C. Douglas, A. Cichocki, and S. Amari, "Multichannel blind separation and deconvolution of sources with arbitrary distributions," in *Neural Networks for Signal Processing VII*, J. Principe, L. Gile, N. Morgan, and E. Wilson, Eds., 1997, pp. 436~445

[21] 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 *Proc. Int. Symp. Nonlinear Theory and Applications*, 1997, pp. 731 ~ 734.

[22] M. Girolami, "An alternative perspective on adaptive independent component analysis algorithms," *Neural Computation*, vol. 10, no. 8, pp. 2103 ~ 2114, Nov. 1998.

[23] 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*, vol. 11, no. 2, pp. 60 9 ~ 663, 1999.

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

[25] J. -F. Cardoso, "Infomax and maximum likelihood for source separation," *IEEE Signal Processing Letters*, vol. 4, no. 4, pp. 112 ~ 114, Apr. 1997.

[26] S. Amari, "Neural gradient for over- and under-complete bases in ICA," *Neural Computation*, vol. 11, no. 8, pp. 1875 ~ 1883, 1999.

[27] I. S. Gradshteyn, I. M. Ryzhik, and A. Jeffrey, *Table of Integrals, Series, and Products*, Academic Press, 1994.

[28] D. J. C. MacKay, "Maximum likelihood and covariant algorithms for independent component analysis," Tech. Rep. Draft 3.7, University of Cambridge, Cavendish Laboratory, 1996.

[29] S. Choi and A. Cichocki, "Adaptive blind separation of speech signals: Cocktail party problem," in *Proc. Int. Conf. Speech Processing*, 1997, pp. 617 ~ 622.

[30] S. Choi, "Nonstationary source separation," *Journal of Electrical Engineering and Information Science*, vol. 4, no. 4, pp. 499 ~ 506, Aug. 1999.

Seungjin Choi was born in Seoul, Korea, on October 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 a 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 probabilistic learning, statistical (blind) signal processing, independent component analysis, multiuser communications, and pattern recognition.