Independent Component Analysis on the Basis of Helmholtz Machine

Masashi OHATA[*1]
ohatama@bmc.riken.go.jp

Toshiharu MUKAI[*1]
tosh@bmc.riken.go.jp

Kiyotoshi MATSUOKA[*2]
matsuoka@brain.kyutech.ac.jp

[*1] Biologically Integrative Sensors Lab., RIKEN BMC Research Center, Japan
[*2] Dep. of Brain Science and Engineering, Kyushu Institute of Technology, Japan

Abstract

This paper addresses an algorithm for independent component analysis on the basis of Helmholtz machine, which is an unsupervised learning machine. The algorithm is constructed by two terms. One is a term for obtaining a desired demixing process, which is the conventional rule on the base of the information theory. The other is a term for evaluating whether the inverse system and the set of obtained components are desirable or not. Due to this term, the algorithm effectively works in blind separation of overdetermined mixture with additive noise. We demonstrate the effectiveness by computer simulation.

Keywords: independent component analysis, blind source separation, Helmholtz machine and online algorithm

1. Introduction

Independent component analysis (ICA) is a statistical technique for decomposing multivariable data into a linear combination of independent components (ICs). ICA is utilized to solve the blind signal separation (BSS) problem.

From a learning-methodological point of view, algorithms for ICA are categorized as unsupervised learning. An other well-known learning method in this class is the wake-sleep algorithm ([7]). It is also known as a learning algorithm associated with the Expectation-Maximization (EM) algorithm ([6]) and it is used as learning of Helmholtz machine proposed by Dayan et. al. [3], [5]. The machine is assembled from the recognition and generative network models. The former network represents a given data to other components on basis of some criterion and the latter network synthesizes the components obtained in the recognition model such that the synthesis equals the given data. In other words, the generative network is designed as an inverse system of the recognition network. Naturally, its learning algorithm has two phases: wake and sleep phases. The wake and sleep phases acquire parameters of the generative and recognition models for the data, respectively.

In general, Helmholtz machine is dealt with as a network whose activation functions are nonlinear. Its special version is a linear network (the activation functions are linear). Neal and Dayan [9] apply the network to factor analysis (FA) model. FA is a statistical technique for obtaining common factors in multivariable data on the base of the second order statistics. When the factors are regarded as ICs, the ICA model has a similar structure to the FA model. The ICA model must be based on the higher order statistic.

In this paper, we deal with the ICA problem for the model that the number of ICs is less than or equal to one of the observations which contain the additive noise. In BSS, it is known as blind separation of overdetermined mixture with additive noise [10]. We propose an algorithm for the ICA problem on the basis of Helmholtz machine in attention to the construction of linear system. Although the machine has two learning phases, the derived learning rule can be expressed as a single rule. Namely, the algorithm is given by the combination of the terms derived from the wake and sleep phases. The term corresponding to the sleep phase is equivalent to one based on 'the inverse minimal distortion principle' proposed by Matsuoka [8]. The obtained algorithm has the robustness against additive noise at the observation.

2. Formulation of ICA

Let us consider that an M-dimensional multivariable data $\mathbf{x}(t) = [x_1(t), ..., x_M(t)]^T$ is observed at discrete time t (the superscript 'T' represents the transpose of a vector or matrix). We assume that the data is generated by the following process

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t) = \sum_{i=1}^{N} s_i(t)\mathbf{a}_i + \mathbf{n}(t), \qquad (1)$$

where $\mathbf{A} = [\mathbf{a}_1, ..., \mathbf{a}_N]$ is an unknown $M \times N$ matrix of full column rank ($M \geq N$) and $\mathbf{a}_i$ ($i = 1, ..., N$) are its column s(which are linearly independent). The elements of $\mathbf{s}(t) = [s_1(t), ..., s_N(t)]^T$ are mutually independent. $\mathbf{n}(t) = [n_1(t), ..., n_M(t)]^T$ is the vector of

additive white Gaussian noise with zero mean. It is independent of $\mathbf{s}(t)$ and each entry of $\mathbf{n}(t)$ is assumed to be sufficiently small as compared with $\{s_i(t)\}$.

Let $p(\mathbf{x})$ and $r(\mathbf{s})$ denote the probability density functions (pdfs) of $\mathbf{x}(t)$ and $\mathbf{s}(t)$, respectively. From the assumption of $\mathbf{s}(t)$, $r(\mathbf{s})$ is expressed by the product of its marginal pdfs $r_i(s_i)$ (i = 1, ..., N): $r(\mathbf{s}) = \prod_{i=1}^{N} r_i(s_i)$. For mathematical simplicity, we assume that the variances of $n_i(t)$ (i =1, ..., N) are same, namely, the pdf of $\mathbf{n}(t)$ is given by

$$p_{\mathbf{n}}(\mathbf{n}) = (2\pi\beta^{-1})^{-M/2}\exp(-\beta\mathbf{n}^T\mathbf{n}/2), \quad (2)$$

where β is a positive value (it represents the inverse of the variance). The joint pdf of $\mathbf{x}(t)$ and $\mathbf{s}(t)$ is expressed as

$$p(\mathbf{x}, \mathbf{s} \mid \mathrm{A}) = p_{\mathbf{n}}(\mathbf{x} - \mathbf{A}\mathbf{s})r(\mathbf{s}). \quad (3)$$

Its marginal pdf of $\mathbf{x}$ is given by.

$$p(\mathbf{x} \mid \mathbf{A}) = \int d\mathbf{s}\, p_{\mathbf{n}}(\mathbf{x} - \mathbf{A}\mathbf{s})r(\mathbf{s}). \quad (4)$$

The goal of the ICA problem is to estimate both $\mathbf{A}$ and $\mathbf{s}(t)$ from $\mathbf{x}(t)$. At least, it is sufficient to estimate $\mathbf{A}$ because the estimate of $\mathbf{s}(t)$ is obtained by using the pseudoinverse of the obtained $\mathbf{A}$ (when $\mathbf{n}(t) = \mathbf{0}$, $\mathbf{A}^{\dagger}\mathbf{x}(t)$ exactly becomes equal to $\mathbf{s}(t)$). Here $\mathbf{A}^{\dagger}$ denotes the pseudoinverse of $\mathbf{A}$. From the viewpoint of the estimation theory, we must obtain $\mathbf{A}$ such that a likelihood function

$$\prod_{t=1}^{T} p(\mathbf{x}(t) \mid \mathbf{A}) \quad (5)$$

takes a maximum (the maximum likelihood method). Instead of (5), we deal with

$$L(\mathbf{A}) = E_{\mathbf{x}}[\log p(\mathbf{x} \mid \mathbf{A})] = \int d\mathbf{x}\, p_{\mathbf{x}}(\mathbf{x}) \log p(\mathbf{x} \mid \mathbf{A}). \quad (6)$$

Maximization of (5) is equivalent to maximize this function. This problem is very difficult because we have the observations $\{\mathbf{x}(t)\}$ only. If both $\{\mathbf{s}(t)\}$ and $\{\mathbf{x}(t)\}$ is given beforehand, the estimation of $\mathbf{A}$ is comparably easy. This case is said to be complete for estimating $\mathbf{A}$. Since $\mathbf{s}(t)$ is unknown in our problem, equation (4) is referred to as 'incomplete data model'. As a method for solving this kind of problem, the EM algorithm is well known. Moreover, as an algorithm related with it, the wake-sleep algorithm is proposed, which is for learning of Helmholtz machine. We deal with the ICA problem on the scheme of the learning machine.

3. Helmholtz machine for ICA

In this section, we describe the maximum likelihood method of (6) in terms of Helmholtz machine.

In order to obtain $\mathbf{A}$ such that $s_1(t)$, ..., $s_N(t)$ are mutually independent, we introduce the following process:

$$\mathbf{s}(t) = \mathbf{W}\mathbf{x}(t) + \mathbf{e}(t), \quad (7)$$

where $\mathbf{W}$ is an $N \times M$ matrix with full of row rank. In BSS, this is called a demixing or separating process. $\mathbf{e}(t) = [e_1(t), ..., e_N(t)]^T$ is the vector of white Gaussian noise with zero mean and covariance matrix $\gamma^{-1}\mathbf{I}$. Equation (7) is the recognition model of the machine. The generative model of the Helmholtz machine is equation (1).

It is explicit that, if $\mathbf{e}(t) = \mathbf{n}(t) = \mathbf{0}$ and $\mathbf{s}(t)$ is a set of ICs, then $\mathbf{W} = \mathbf{A}^{\dagger}$ or $\mathbf{A} = \mathbf{W}^{\dagger}$ is a solution of this problem.

Let $q_{\mathbf{e}}(\mathbf{e})$ be the pdf of $\mathbf{e}(t)$, it is given by

$$q_{\mathbf{e}}(\mathbf{e}) = (2\pi\gamma^{-1})^{-N/2}\exp(-\gamma\mathbf{e}^T\mathbf{e}/2). \quad (8)$$

Here we assume that γ^{-1} is sufficiently small. The conditional pdf of $\mathbf{s}$ when $\mathbf{x}$ is given is expressed by

$$q(\mathbf{s} \mid \mathbf{x}) = q_{\mathbf{e}}(\mathbf{s} - \mathbf{W}\mathbf{x}), \quad (9)$$

and the joint pdf of $(\mathbf{x}, \mathbf{s})$ is given as

$$q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}) = q_{\mathbf{e}}(\mathbf{s} - \mathbf{W}\mathbf{x})p(\mathbf{x}). \quad (10)$$

From equations (3) and (4), $\log p(\mathbf{x} \mid \mathbf{A})$ is expressed by

$$\log p(\mathbf{x} \mid \mathbf{A}) = \log \frac{p(\mathbf{x}, \mathbf{s} \mid \mathbf{A})}{p(\mathbf{s} \mid \mathbf{x}, \mathbf{A})}.$$

By using (9) and this, $L(\mathbf{A})$ is reexpressed as

$$\begin{aligned} L(\mathbf{A}) &= -\int d\mathbf{x} d\mathbf{s}\, q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}) \log p(\mathbf{x} \mid \mathbf{A}) \\ &= -E_{\mathbf{x}}[F(\mathbf{A} \mid \mathbf{W}, \mathbf{x})] \\ &\quad + D[q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}), p(\mathbf{x}, \mathbf{s} \mid \mathbf{A})], \end{aligned} \quad (11)$$

where $F(\mathbf{A} \mid q, \mathbf{x})$ is given by

$$\begin{aligned} F(\mathbf{A} \mid \mathbf{W}, \mathbf{x}) \triangleq & -\int d\mathbf{s}\, q(\mathbf{s} \mid \mathbf{x}) \log r(\mathbf{s})\, p_{\mathbf{n}}(\mathbf{x} - \mathbf{A}\mathbf{s}) \\ & + \int d\mathbf{s}\, q(\mathbf{s} \mid \mathbf{x}) \log q(\mathbf{s} \mid \mathbf{x}), \end{aligned} \quad (12)$$

and $D[q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}), p(\mathbf{x}, \mathbf{s} \mid \mathbf{A})]$ denotes the Kulback-Leibler divergence (KL-divergence) between $q(\mathbf{x}, \mathbf{s} \mid \mathbf{W})$ and $p(\mathbf{x}, \mathbf{s} \mid \mathbf{A})$ and is defined by

$$\begin{aligned} & D[q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}), p(\mathbf{x}, \mathbf{s} \mid \mathbf{A})] \\ &= \int d\mathbf{s} d\mathbf{x} q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}) \log \frac{q(\mathbf{x}, \mathbf{s} \mid \mathbf{W})}{p(\mathbf{x}, \mathbf{s} \mid \mathbf{A})} \end{aligned} \quad (13)$$

It takes non-negative value, and it takes a minimum, 0 if and only if $q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}) = p(\mathbf{x}, \mathbf{s} \mid \mathbf{A})$ holds.

Let $p_{\mathbf{y}}(\mathbf{y})$ be the pdf of $\mathbf{y}(t) = \mathbf{W}\mathbf{x}(t)$. Note that the parameter of $p_{\mathbf{y}}(\mathbf{y})$ is $\mathbf{W}$. By using this pdf, equation (13) is taken apart as

$$\begin{aligned} & D[q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}), p(\mathbf{x}, \mathbf{s} \mid \mathbf{A})] \\ &= D[q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}), p_{\mathbf{n}}(\mathbf{x} - \mathbf{A}\mathbf{s})p_{\mathbf{y}}(\mathbf{s})] \\ &\quad + \int d\mathbf{s} d\mathbf{x} q(\mathbf{s} - \mathbf{W}\mathbf{x})\, p(\mathbf{x}) \log \frac{p_{\mathbf{n}}(\mathbf{x} - \mathbf{A}\mathbf{s})\, p_{\mathbf{y}}(\mathbf{s})}{p(\mathbf{x}, \mathbf{s} \mid \mathbf{A})} \end{aligned} \quad (14)$$

Since γ^{-1} is sufficiently small, the second term is approximated to $D[p_{\mathbf{y}}(\mathbf{y}), r(\mathbf{y})]$ and it is the evaluation function in the conventional ICA method. Therefore, we have

$$\begin{aligned} & D[q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}), p(\mathbf{x}, \mathbf{s} \mid \mathbf{A})] \\ &\approx D[q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}), p_{\mathbf{n}}(\mathbf{x} - \mathbf{A}\mathbf{s})p_{\mathbf{y}}(\mathbf{s})] \\ &\quad + D[p_{\mathbf{y}}(\mathbf{y}), r(\mathbf{y})]. \end{aligned} \quad (15)$$

Since KL-divergence is larger than or equal to 0, we have

$$\begin{aligned} L(\mathbf{A}) \approx & -E_{\mathbf{x}}[F(\mathbf{A} \mid \mathbf{W}, \mathbf{x})] + D[p_{\mathbf{y}}(\mathbf{y}), r(\mathbf{y})] \\ & + D[q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}), p_{\mathbf{n}}(\mathbf{x} - \mathbf{A}\mathbf{s})p_{\mathbf{y}}(\mathbf{s})] \end{aligned}$$

$$\geq -E_{\mathbf{x}}[F(\mathbf{A} \mid \mathbf{W}, \mathbf{x})] + D[p_{\mathbf{y}}(\mathbf{y}), r(\mathbf{y})]. \quad (16)$$

$D[q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}), p_{\mathbf{n}}(\mathbf{x}-\mathbf{A}\mathbf{s})p_{\mathbf{y}}(\mathbf{s})]$ takes a minimum, 0, for any pairs of $(\mathbf{A}, \mathbf{W})$ such that $\mathbf{W} = \mathbf{A}^{\dagger}$ holds. In the ICA problem, this divergence is unmeaning. Since $D[p_{\mathbf{y}}(\mathbf{y}), r(\mathbf{y})]$ is also non-negative, $L(\mathbf{A})$ is bounded as

$$L(\mathbf{A}) \gtrsim -E_{\mathbf{x}}[F(\mathbf{A} \mid \mathbf{W}, \mathbf{x})].$$

If $\mathbf{n}(t) = \mathbf{0}$, then we can obtain $\mathbf{W}$ such that the elements of $\mathbf{y}(t)$ become independent components $\{s_i(t)\}$, by minimizing $D[p_{\mathbf{y}}(\mathbf{y}), r(\mathbf{y})]$ with respect to $\mathbf{W}$. In the case where $\mathbf{n}(t)$ is sufficiently small, we can also obtain a desired demixing process by minimizing the divergence. After that, by maximizing $-E_{\mathbf{x}}[F(\mathbf{A} \mid \mathbf{W}, \mathbf{x})]$ with respect to $\mathbf{A}$ for obtained $\mathbf{W}$, $L(\mathbf{A})$ is indirectly maximized [at the same time, $D[q(\mathbf{x}, \mathbf{s}| \mathbf{W}), p_{\mathbf{n}}(\mathbf{x}-\mathbf{A}\mathbf{s})p_{\mathbf{y}}(\mathbf{s})]$ is minimized].

We summarize the above as below.

Recognition model (R-model)

The recognition model is

$$\mathbf{s}(t) = \mathbf{W}\mathbf{x}(t) + \mathbf{e}(t).$$

The joint pdf of $(\mathbf{x}(t), \mathbf{s}(t))$ is expressed by

$$q(\mathbf{x}, \mathbf{s} \mid \mathbf{W}) = q_{\mathbf{e}}(\mathbf{s} - \mathbf{W}\mathbf{x})p(\mathbf{x}).$$

The parameter of this model is an $N \times M$ matrix $\mathbf{W}$, which has full row rank. The cost function for obtaining $\mathbf{W}$ is

$$D[p_{\mathbf{y}}(\mathbf{y}), r(\mathbf{y})] = \int d\mathbf{y} p_{\mathbf{y}}(\mathbf{y}) \log \frac{p_{\mathbf{y}}(\mathbf{y})}{r(\mathbf{y})} \triangleq R(\mathbf{W}|\mathbf{A}). \quad (17)$$

This is the same as the cost function of the conventional method [3].

Generative model (G-model)

The generative model is

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t).$$

The joint pdf of $(\mathbf{x}(t), \mathbf{s}(t))$ is expressed by

$$p(\mathbf{x}, \mathbf{s} \mid \mathrm{A}) = p_{\mathbf{n}}(\mathbf{x} - \mathbf{A}\mathbf{s})r(\mathbf{s}).$$

The parameter of this is an $M \times N$ matrix $\mathbf{A}$ with being full of column rank. The cost function is given by

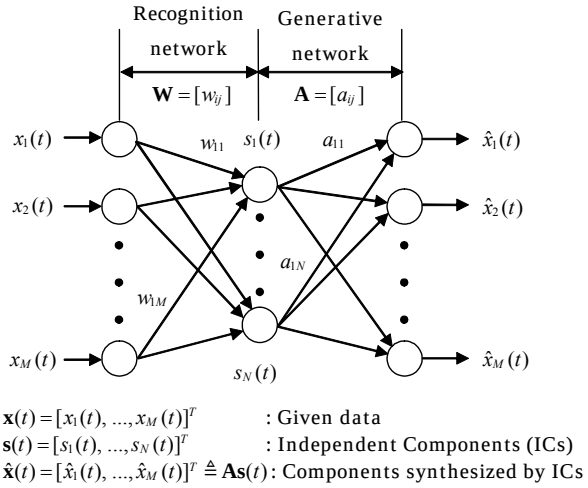
$$\begin{aligned} G(\mathbf{A} \mid \mathbf{W}) &\triangleq E_{\mathbf{x}}[F(\mathbf{A} \mid \mathbf{W}, \mathbf{x})] \\ &= -\int d\mathbf{x} d\mathbf{s}\, q(\mathbf{s} \mid \mathbf{x})\, p(\mathbf{x}) \log r(\mathbf{s})\, p_{\mathbf{n}}(\mathbf{x}-\mathbf{A}\mathbf{s}) \\ &\quad + \int d\mathbf{x} d\mathbf{s}\, q(\mathbf{s} \mid \mathbf{x})\, p(\mathbf{x}) \log q(\mathbf{s} \mid \mathbf{x}). \end{aligned} \quad (18)$$

Substituting (9) into this and calculating the expectation of this with respect to $p_{\mathbf{x}}(\mathbf{x})$, we have

$$G(\mathbf{A} \mid \mathbf{W}) \approx -E_{\mathbf{x}, \mathbf{y}}[\log p_{\mathbf{n}}(\mathbf{x} - \mathbf{A}\mathbf{y})] + const. \quad (19)$$

The derivation from (18) to (19) is shown in Appendix. In this step, it is sufficient to minimize $-E_{\mathbf{x}, \mathbf{y}}[\log p_{\mathbf{n}}(\mathbf{x}-\mathbf{A}\mathbf{y})]$ with respect to $\mathbf{A}$. The minimization is equivalent to that of $E_{\mathbf{x}, \mathbf{y}}[\|\mathbf{x} - \mathbf{A}\mathbf{y}\|^2]$ because $p_{\mathbf{n}}(\mathbf{n})$ is the pdf of multivariable Gaussian distribution with the covariance matrix $\beta^{-1}\mathbf{I}$. This is a meaningful part in the cost function $G(\mathbf{A} \mid \mathbf{W})$.

The structure of the Helmholtz machine for ICA is illustrated in Figure 1. This shows the case of noise free ($\mathbf{n}(t) = \mathbf{0}$).



$\mathbf{x}(t) = [x_1(t), ..., x_M(t)]^T$: Given data
$\mathbf{s}(t) = [s_1(t), ..., s_N(t)]^T$: Independent Components (ICs)
$\hat{\mathbf{x}}(t) = [\hat{x}_1(t), ..., \hat{x}_M(t)]^T \triangleq \mathbf{A}\mathbf{s}(t)$: Components synthesized by ICs

Figure 1: Helmholtz machine for ICA in the case of noise free.

The learning procedure of the Helmholtz machine for ICA is summarized as follows.

(i) Wake phase: the recognition model

Minimize $D[p_{\mathbf{y}}(\mathbf{y}), r(\mathbf{y})]$ with respect to $\mathbf{W}$. Note that the minimization is independent of $\mathbf{A}$.

(ii) Sleep phase: the generative model

For $\mathbf{W}$ obtained in the wake phase, minimize $-E_{\mathbf{x}, \mathbf{y}}[\log p_{\mathbf{n}}(\mathbf{x}-\mathbf{A}\mathbf{y})]$ with respect to $\mathbf{A}$.

The two steps are repeated until the difference between the value of $E_{\mathbf{x}}[F(\mathbf{A} \mid \mathbf{W}, \mathbf{x})]$ and its previous value becomes sufficiently small.

4. An Online Algorithm

For the minimizations in both the wake and sleep phases, we use the natural gradient method proposed by Amari [1]. The article describes the natural gradient method on the manifold formed by nonsingular squared matrices. The method is easily expanded to the case of those formed by rectangular matrices with full rank. Let $\mathcal{M}_{N\times M}$ ($\mathcal{M}_{M\times N}$) be the manifold whose elements are given by $N \times M$ ($M \times N$) matrices with full rank. $\mathbf{W}$ or $\mathbf{A}$ is a point of each manifold. We consider the natural gradient method on $\mathcal{M}_{N\times M}$ ($\mathcal{M}_{M\times N}$) for the wake (sleep) phase. Let $d\mathbf{W}$ ($d\mathbf{A}$) be a tangent vector of the manifold $\mathcal{M}_{N\times M}$ ($\mathcal{M}_{M\times N}$) at $\mathbf{W}$ ($\mathbf{A}$). We introduce the following Riemannian metrics, namely, norms, for the tangent vectors $d\mathbf{W}$ and $d\mathbf{A}$ as

$$\|d\mathbf{W}\|_{\mathbf{W}} = \|d\mathbf{W}\mathbf{W}^{\dagger}\| \text{ for } \mathcal{M}_{N\times M},$$

and

$\|d\mathbf{A}\|_{\mathbf{A}} = \|d\mathbf{A}\mathbf{A}^{\dagger}\|$ for $\mathcal{M}_{M\times N}$,

where $\|\cdot\|$ denotes the Frobenius norm of a matrix. Note that $d\mathbf{W}$ must be constrained on the space spanned by the row vectors of $\mathbf{W}$.

By obtaining the optimal directions of $d\mathbf{W}$ and $d\mathbf{A}$ under the constraints that the norms are constant, we have the natural gradient algorithms for the wake and sleep phases, respectively. We omit the derivations.

Learning rule for the R-model (wake phase)

The natural gradient of $R(\mathbf{W} \mid \mathbf{A})$ with respect to $\mathbf{W}$ is calculated as

$$\frac{\partial R(\mathbf{W} \mid \mathbf{A})}{\partial \mathbf{W}} \mathbf{W}^T\mathbf{W} = -(\mathbf{I} - E_{\mathbf{y}}[\varphi(\mathbf{y})\mathbf{y}^T])\mathbf{W} \triangleq g_1(\mathbf{W}), \tag{20}$$

where $\varphi(\mathbf{y}) \triangleq [\varphi_1(y_1), \ldots, \varphi_N(y_N)]^T$ and $\varphi_i(u) \triangleq -\dot{r}_i(u)/r_i(u)$ ($i = 1, \ldots, N$). Thus, the learning rule for minimizing $R(\mathbf{W} \mid \mathbf{A})$ is given by

$$\frac{d\mathbf{W}}{du} = -g_1(\mathbf{W}) = (\mathbf{I} - E_{\mathbf{y}}[\varphi(\mathbf{y})\mathbf{y}^T])\mathbf{W}. \tag{21}$$

Note that the initial value of $\mathbf{W}$ is set as being full of row rank. This is equivalent to the algorithm proposed by Bell and Sejnowski [3]. The equilibrium condition of (21) is given by

$$\mathbf{I} - E_{\mathbf{y}}[\varphi(\mathbf{y})\mathbf{y}^T] = \mathbf{O} \tag{22}$$

This gives N^2 constraints to $\mathbf{W}$. This states that there are $N(M-N)$ degrees of freedom in $\mathbf{W}$ obtained by this dynamics.

Learning rule for the G-model (sleep-phase)

In the same way as the algorithm for the recognition model, the natural gradient algorithm for the sleep phase is given by

$$\frac{d\mathbf{A}}{du} = \beta E_{\mathbf{x},\mathbf{y}}[(\mathbf{x} - \mathbf{A}\mathbf{y})\mathbf{y}^T]\mathbf{A}^T\mathbf{A}. \tag{23}$$

This dynamics estimates the mixing process $\mathbf{A}$ when $y_i(t)$ ($i = 1, \ldots, N$) are mutually independent. The demixing process $\mathbf{W}$ must be uniquely determined for $\mathbf{A}$, namely, some relationship between them need hold. In our case, the relationship is given by its pseudoinverse:

$$\mathbf{W} = \mathbf{A}^{\dagger} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \text{ or } \mathbf{A} = \mathbf{W}^T(\mathbf{W}\mathbf{W}^T)^{-1}. \tag{24}$$

Thus, it is conventional to express the dynamics in the form of $\mathbf{W}$. When $\mathbf{A}$ changes from $\mathbf{A}$ to $\mathbf{A} + d\mathbf{A}$, the deviation $d\mathbf{W}$ is given by

$$d\mathbf{W} = \mathbf{W}\mathbf{W}^T d\mathbf{A}^T\{\mathbf{I} - \mathbf{W}^T(\mathbf{W}\mathbf{W}^T)^{-1}\mathbf{W}\} - \mathbf{W}d\mathbf{A}\mathbf{W}. \tag{25}$$

Thus, algorithm (23) can be expressed in the form of $\mathbf{W}$. From (24) and (25), we have

$$\frac{d\mathbf{W}}{du} = \beta E_{\mathbf{x},\mathbf{y}}[\mathbf{y}\mathbf{x}^T - \mathbf{y}\mathbf{y}^T(\mathbf{W}\mathbf{W}^T)^{-1}\mathbf{W}] \triangleq g_2(\mathbf{W}). \tag{26}$$

This can provide the $N(M-N)$ remained constraints for the demixing process $\mathbf{W}$ obtained by (21). We discuss the detail of this in Section 6.

Learning algorithm for ICA

Combining equations (21) and (26), we have

$$\begin{aligned}\frac{d\mathbf{W}}{du} &= g_1(\mathbf{W}) + g_2(\mathbf{W}) \\ &= (\mathbf{I} - E_{\mathbf{y}}[\varphi(\mathbf{y})\mathbf{y}^T])\mathbf{W} \\ &\quad + \beta E_{\mathbf{x},\mathbf{y}}[\mathbf{y}\mathbf{x}^T - \mathbf{y}\mathbf{y}^T(\mathbf{W}\mathbf{W}^T)^{-1}\mathbf{W}].\end{aligned} \tag{27}$$

Replacing the expectations with their instantaneous value and the differential operation with the difference operation Δ (namely, $\Delta\mathbf{W}$ denotes the updated values of $\mathbf{W}$), we can derive the stochastic version of algorithm (27) as follows:

$$\begin{aligned}\Delta\mathbf{W} = \alpha\{&(\mathbf{I} - \varphi(\mathbf{y}(t))\mathbf{y}^T(t))\mathbf{W} \\ &+ \beta(\mathbf{y}(t)\mathbf{x}^T(t) - \mathbf{y}(t)\mathbf{y}^T(t)(\mathbf{W}\mathbf{W}^T)^{-1}\mathbf{W})\},\end{aligned} \tag{28}$$

where α is a learning rate. $(\mathbf{W}\mathbf{W}^T)^{-1}$ is also obtained recursively as follows.

Setting $\mathbf{V} = \mathbf{W}\mathbf{W}^T$ and letting $\bar{\mathbf{V}}^{(k)}$ be the k-th recursive value of $\mathbf{V}^{-1}$, we obtain $\mathbf{V}^{-1} = (\mathbf{W}\mathbf{W}^T)^{-1}$ in a small number of steps, by

$$\bar{\mathbf{V}}^{(k)} = 2\bar{\mathbf{V}}^{(k-1)} - \bar{\mathbf{V}}^{(k-1)}\mathbf{V}\bar{\mathbf{V}}^{(k-1)}.$$

Note that this recursive rule must be executed at each update of $\mathbf{W}$.

5. Example

We demonstrate the validity of our algorithm by computer simulation. Let us consider the case where $N = 2$, $M = 3$ and $\mathbf{n}(t) = \mathbf{0}$ in the mixing process (1), that is,

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \mathbf{A}\mathbf{s}(t) = \mathbf{A}\begin{bmatrix} s_1(t) \\ s_2(t) \end{bmatrix}.$$

In this equation, we set $\mathbf{A}$ to

$$\mathbf{A} = \begin{bmatrix} 0.75 & 0.35 \\ -0.36 & -0.86 \\ -0.45 & -0.61 \end{bmatrix}$$

Each of $s_i(t)$ ($i = 1, 2$) was the signal taking 4 values $\{-3, -1, 1, 3\}$ with equal probabilities. Since the signals are sub-Gaussian, we set $\varphi_i(u) = u^3$ (according to [1], [3]), $\alpha = 1.0\times10^{-4}$ and $\beta = 0.5$. Figures 2 and 3 show the mixed source signals and the obtained components by the learning rule (28), respectively. Figure 3 says that $g_2(\mathbf{W})$ in (28) does not cause a problem and seems not to play any role. In the case where the observation $\mathbf{x}(t)$ involves additive noise, $g_2(\mathbf{W})$ plays the important role. We argue this case in the next section.

6. Discussion

If the number of ICs equals that of the observed signals ($N = M$), the relationship between $\mathbf{A}$ and $\mathbf{W}$

is given as $\mathbf{W} = \mathbf{A}^{-1}$. Its differential of $\mathbf{W}$ is expressed as $d\mathbf{W} = -\mathbf{W}d\mathbf{A}\mathbf{W}$ and then $g_2(\mathbf{W})$ becomes zero. Accordingly, in this case, the learning rule is the same algorithm as Bell and Sejnowski [3] proposes:

$$\Delta\mathbf{W} = \alpha[\mathbf{I} - \varphi(\mathbf{y}(t))\mathbf{y}^T(t)]\mathbf{W}. \quad (29)$$

Let $\mathbf{W}^*$ be a matrix which satisfies the condition (22). Hereafter we call the matrix a desired demixing process. In the case that $M > N$ and $\mathbf{n}(t) = \mathbf{0}$, $\mathbf{W}^*$ is expressed by

$$\mathbf{W}^* = \mathbf{D}\mathbf{A}^{\dagger} + \mathbf{Z}, \quad (30)$$

where $\mathbf{D}$ is an $N \times N$ nonsingular diagonal matrix, $\mathbf{Z}$ an $N \times M$ matrix which satisfies $\mathbf{Z}\mathbf{A} = \mathbf{O}$. While $\mathbf{D}$ is determined by the equilibrium condition, $\mathbf{Z}$ is not uniquely obtained by (29) only. The rows of $\mathbf{Z}$ are orthogonal to the hyperplane spanned by $\{\mathbf{a}_i\}$. Thus, the degree of freedom in $\mathbf{W}^*$ of (30) is $N(M-N)$.

The equilibrium condition of (26) is given as

$$g_2(\mathbf{W}) = E_{\mathbf{x},\mathbf{y}}[\mathbf{y}\mathbf{x}^T - \mathbf{y}\mathbf{y}^T(\mathbf{W}\mathbf{W}^T)^{-1}\mathbf{W}] = \mathbf{O}.$$

In the set of the desired demixing processes in the form of (30), an element which satisfies this condition is given by

$$\mathbf{W}^* = \mathbf{D}\mathbf{A}^{\dagger}. \quad (31)$$

Accordingly, (26) plays the role of searching for this demixing process in the set.

Let $\mathbf{y}^*(t)$ denote the output signal of the mixing process $\mathbf{W}^*$. It is expressed as

$$\mathbf{y}^*(t) = \mathbf{D}\mathbf{s}(t) + (\mathbf{D}\mathbf{A}^{\dagger} + \mathbf{Z})\mathbf{n}(t) = \mathbf{y}_s^*(t) + \mathbf{y}_n^*(t) \quad (32)$$

Here $\mathbf{y}_s^*(t) = [y_{s1}^*(t), ..., y_{sN}^*(t)]^T$ and $\mathbf{y}_n^*(t) = [y_{n1}^*(t), ..., y_{nN}^*(t)]^T$ are defined as $\mathbf{y}_s^*(t) \triangleq \mathbf{D}\mathbf{s}(t)$ and $\mathbf{y}_n^*(t) \triangleq (\mathbf{D}\mathbf{A}^{\dagger} + \mathbf{Z})\mathbf{n}(t)$, respectively. The variance of $\mathbf{y}_n^*(t)$ is calculated as

$$E[(\mathbf{y}_n^*(t))^T \mathbf{y}_n^*(t)] = \beta^{-1}\mathrm{tr}\{(\mathbf{D}\mathbf{A}^{\dagger} + \mathbf{Z})(\mathbf{D}\mathbf{A}^{\dagger} + \mathbf{Z})^T\}.$$

Since $\mathbf{Z}$ must satisfy $\mathbf{Z}\mathbf{A} = \mathbf{O}$, this is minimum if and only if $\mathbf{Z} = \mathbf{O}$. Accordingly, algorithm (28) can be used to search for a desired separator which suppresses $\mathbf{y}_n^*(t)$ as small as possible. When the mixing process is given by (1), the desired demixing process (31) suppresses the influence of $\mathbf{n}(t)$ in the output, namely, the algorithm has robustness against noise $\mathbf{n}(t)$.

By computer simulation, we demonstrate the effectiveness of the second term of (28). We define the signal-to-noise ratio (SNR) of $\mathbf{y}^*(t)$ as

$$\mathrm{SNR} \triangleq \sum_{i=1}^{N} \sigma_i^s / \sigma_i^n / N, \quad (33)$$

where σ_i^s and σ_i^n denote the standard deviations of $y_{si}^*(t)$ and $y_{ni}^*(t)$, respectively.

The conditions on ICs and $\mathbf{A}$ are the same as shown in Example. Each entry of $\mathbf{n}(t)$ is assumed to obey the Gaussian distribution with zero mean and 0.01 variance.

Set $\varphi_i(u) = u^3$ ($i = 1, 2$) and the initial value of $\mathbf{W}$ to

$$\mathbf{W} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

For algorithm (28) with $\beta = 0.5$ and the learning rule (29), we executed the algorithms and then obtained the values of SNR as shown in the following table [although, in the both cases, ICs were obtained, we omit illustrating the waveforms of the outputs and their distribution].

From the result, algorithm (28) is judged to be better than algorithm (29). Consequently, the new rule has robustness against noise as compared with the conventional algorithm. The robustness results from the new term, namely, $g_2(\mathbf{W})$.

Table 1 Values of SNR

The algorithm	(28)	(29)
SNR	32.6	25.0

7. Conclusion

This paper derived a learning rule for ICA from the scheme of Helmholtz machine. The algorithm involves the conventional algorithm proposed by Bell and Sejnowski [3] and the new term. The new term restrains the reflection of the additive noise as small as possible in the obtained ICs. Our algorithm is effective for blind separation of overdetermined mixture with additive noise.

It is the same term as derived from 'inverse minimal distortion principle' (IMDP) proposed by Matsuoka [8]. Conversely, it is considered that IMDP plays the role of the sleep phase of the Helmholtz machine.

If $\mathbf{A}$ is a nonsingular squared matrix, then $g_2(\mathbf{W}) = \mathbf{O}$ and the learning rule becomes the conventional algorithm. Thus, this scheme is reasonable.

Reference

[1] S. Amari, T. P. Chen, and A. Cichocki, "Stability analysis of adaptive blind source separation," Neural Networks, Vol.10, No.8, pp.1355-1361, 1997.

[2] S. Amari, "Natural gradient learning works efficiently in learning," Neural Computation, Vol.10, No.2, pp. 251-276, 1998.

[3] J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," Neural Computation, Vol.7, No.6, pp.1129-1159, 1995.

[4] P. Dayan, G. E. Hinton, R.M. Neal, and R.S. Zemel, "The Helmholtz machine," Neural Computation, Vol.7, No.5, pp.867-888, 1995.

[5] P. Dayan, P. and G. E. Hinton, "Varieties of

Helmholtz machine," Neural Networks, No.9, pp. 1385- 1403, 1996.
[6] A. P. Dempster, N. M. Laird and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," J. Royal Statist. Soc., B Vol. 39, pp.1-38, 1977.
[7] G. E. Hiton, P. Dayan, B. J. Fery and R. M. Neal, "The wake-sleep algorithm for unsupervised neural networks," Science, 268, pp.1158-1161, 1995.
[8] K. Matsuoka, "Principles for eliminating two kinds of indeterminacy in blind source separation," In Proc. of International Conference on Digital Signal Processing, pp. 147-150, 2002.
[9] R. M. Neal and P. Dayan, "Factor analysis using delta-rule wake-sleep learning," Neural Computation, Vol. 9, No. 8, pp.1781-1803, 1997.
[10] L.Q. Zhang, A. Cichocki and S. Amari, "Natural gradient algorithm for blind source separation of overdetermined mixture with additive noise," IEEE Signal processing letters, Vol.6, No.11, November, pp.293-295, 1999.

Appendix: The derivation of (19)

Substituting (9) into (18), we have

$$E_{\mathbf{x}}[F(\mathbf{A}|\mathbf{W}, \mathbf{x})] = -E_{\mathbf{x}}[\int d\mathbf{s} q_{\mathbf{e}}(\mathbf{s}-\mathbf{W}\mathbf{x}) \log p_{\mathbf{n}}(\mathbf{x}-\mathbf{A}\mathbf{s})] - E_{\mathbf{x}}[\int d\mathbf{s} q_{\mathbf{e}}(\mathbf{s}-\mathbf{W}\mathbf{x}) \log r(\mathbf{s})] + \int d\mathbf{x} p_{\mathbf{x}}(\mathbf{x}) \int d\mathbf{s} q_{\mathbf{e}}(\mathbf{s}-\mathbf{W}\mathbf{x}) \log q_{\mathbf{e}}(\mathbf{s}-\mathbf{W}\mathbf{x}).$$

The third term is the negative entropy of $q_{\mathbf{e}}(\mathbf{e})$ and is independent of $\mathbf{A}$. Note that the entropy is invariant against the translation of the variable. Since the variance of $q_{\mathbf{e}}(\mathbf{e})$ is sufficiently small, it is approximate to the Dirac delta function $\delta(\|\mathbf{e}\|)$. Accordingly, the first and second terms can be replaced with $-E_{\mathbf{x},\mathbf{y}}[\log p_{\mathbf{n}}(\mathbf{x}-\mathbf{A}\mathbf{y})]$ and $-E_{\mathbf{y}}[\log r(\mathbf{y})]$, respectively. Thus, we have

$$E_{\mathbf{x}}[F(\mathbf{A}|\mathbf{W}, \mathbf{x})] \approx -E_{\mathbf{x},\mathbf{y}}[\log p_{\mathbf{n}}(\mathbf{x}-\mathbf{A}\mathbf{y})] - E_{\mathbf{y}}[\log r(\mathbf{y})] - H[q_{\mathbf{e}}(\mathbf{e})],$$

where $H[q_{\mathbf{e}}(\mathbf{e})]$ denotes the entropy of $p_{\mathbf{e}}(\mathbf{e})$. Since the second and third terms do not involve $\mathbf{A}$, the minimization of $E_{\mathbf{x}}[F(\mathbf{A}|\mathbf{W}, \mathbf{x})]$ is equivalent to that of $-E_{\mathbf{x},\mathbf{y}}[\log p_{\mathbf{n}}(\mathbf{x}-\mathbf{A}\mathbf{y})]$.

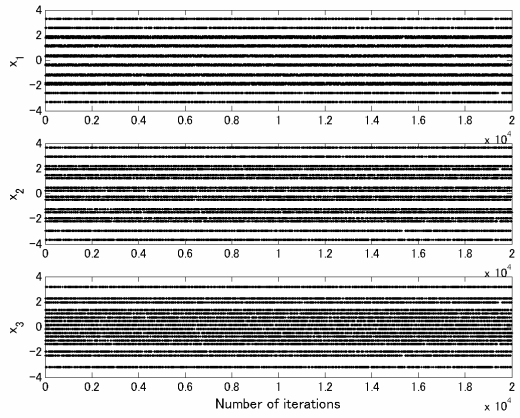
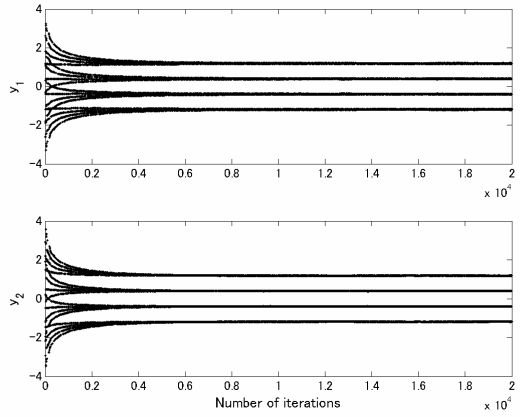
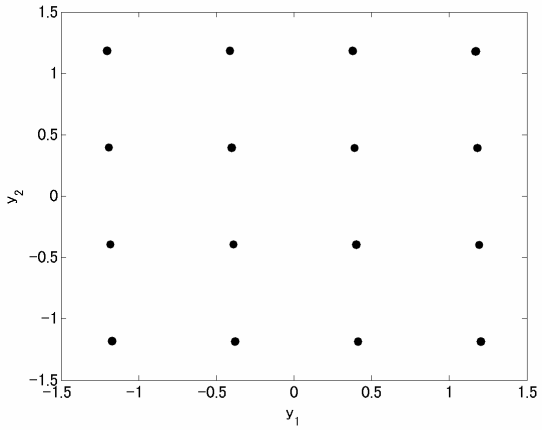


Figure 2: Observed signals



(a) Time transition of output signals



(b) Distribution of output signals

Figure 3: Output signals of $\mathbf{W}$