24
논문 98-01-05

Neural Learning Algorithms for Independent Component Analysis

Seung-Jin Choi*
(崔 丞 鎭*)

Abstract

Independent Component analysis (ICA) is a new statistical method for extracting statistically independent components from their linear instantaneous mixtures which are generated by an unknown linear generative model. The recognition model is learned in unsupervised manner so that the recovered signals by the recognition model become the possibly scaled estimates of original source signals.

This paper addresses the neural learning approach to ICA. As recognition models a linear feedforward network and a linear feedback network are considered. Associated learning algorithms for both networks are derived from maximum likelihood and information-theoretic approaches, using natural Riemannian gradient [1]. Theoretical results are confirmed by extensive computer simulations.

Indexing Terms: Independent component analysis, neural networks, natural Riemannian gradient, unsupervised learning.

I. Introduction[1]

Independent component analysis is a fundamental statistical method encountered in many applications such as feature extraction, digital communications, robust speech recognition, image processing, and biomedical signal analysis (like ECG, EEG, and MEG). In many applications, the sensory signals (observation obtained from multiple sensors) are generated by an (unknown) linear generative model. In other words, observations are linear instantaneous mixtures of unknown source signals. It is desirable to recover the source signals from observations by building the recognition model.

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

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

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

The task of ICA is to recover source signals $\boldsymbol{s}(t)$ from the observations $\boldsymbol{x}(t)$ without the knowledge of $\boldsymbol{A}$ nor $\boldsymbol{s}(t)$. This is often called as blind source separation (BSS). We build a recognition model which transforms the observations $\boldsymbol{x}(t)$ to the network output

* 忠北大學校 電氣電子工學部
(School of Electrical and Electronics Engineering, Chungbuk National University)
接受日: 1998年4月2日, 修正完了日: 1998年7月16日

signals $\boldsymbol{y}(t)$ whose elements are statistically mutually independent, so that the output signals $\boldsymbol{y}(t)$ are possibly scaled estimates of source signals $\boldsymbol{s}(t)$. Inherently there are two indeterminacies in ICA [2]: (1) scaling ambiguity; (2) permutation ambiguity. That is, the recovered signals $\boldsymbol{y}(t)$ by a recognition model are $\boldsymbol{y}(t) = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{s}(t)$, where $\boldsymbol{P}$ is a permutation matrix and $\boldsymbol{\Lambda}$ is some nonsingular diagonal matrix.

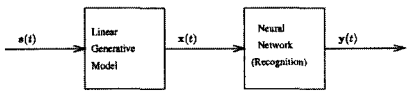

Figure 1: The observation $\boldsymbol{x}(t)$ are generated from unknown source signals $\boldsymbol{s}(t)$ via a linear mapping $\boldsymbol{A}$. The source signals $\boldsymbol{s}(t)$ and a linear mapping in a linear generative models are unknown. The unknown source signals are recovered by neural networks (recognition models) using associated unsupervised learning algorithms.

Since Jutten and Herault [3] proposed a linear feedback network with a simple unsupervised learning algorithm, several methods have been developed. Cichocki *et al.* [4] further developed Jutten and Herault's network by introducing self-normalization connections. Comon [2] gave a good insight to ICA problem from the statistical point of view. Bell and Sejnowski [5] adopted an information maximization principle to find a solution to ICA problem. Maximum likelihood estimation [6] was proposed by Pham *et al.* and was elaborated by MacKay [7]. Serial updating rule was introduced by Cardoso and Laheld [8] and the resulting algorithm was shown to have equivariant performance. Independently, natural gradient was proposed and applied to ICA by Amari *et al.* [9]. Conditions on cross-cumulants for the separation of source signals were investigated by Choi *et al.* [10], Choi and Cichocki [11].

In this paper, we first review the maximum likelihood and information-theoretic approaches. Then, using the natural Riemannian gradient, neural learning algorithms for ICA are derived for a linear feedforward network and a linear feedback network. This paper is organized as follows. In Section 2, maximum likelihood approach to ICA is described. In Section 3 the minimization of mutual information for ICA is discussed. Section 4 provides neural network models for recognition and presents associated ICA algorithms using natural Riemannian gradient. Several discussions are given in Section 5. Computer simulation results are provided in Section 6. Section 7 concludes this paper.

Ⅱ. Maximum Likelihood

Many statistical models are generative models which make use of latent variables to describe a probability distribution over observations. In the context of ICA, the source signals $\boldsymbol{s}(t)$ can be viewed as latent variables which are not directly observable to us, but are observed through the sensor output signals $\boldsymbol{x}(t)$.

The observations $\boldsymbol{x}(t)$ are assumed to be generated from latent variables $\boldsymbol{s}(t)$ (source signals in ICA) via a linear mapping $\boldsymbol{A}$. The mathematical model is given in (1). For the sake of simplicity, we assume that we have as many sensors as source signals, i.e., $m = n$. The extension to the case where we have more sensors than sources will be explained in Section 5.

Consider a set of T independent observations, $X = \{\boldsymbol{x}(t)\}_{t=1}^{T}$. Source signals $\boldsymbol{s}(t)$ are assumed to be statistically independent and their probability density function is denoted by $r(\cdot)$. The likelihood function is given by

$$p(X \mid \boldsymbol{A}, r) = \prod_{t=1}^{T} p(\boldsymbol{x}(t) \mid \boldsymbol{A}, r). \tag{2}$$

The (unknown) mixing matrix $\boldsymbol{A}$ can be learned by

maximizing the likelihood function (2).

A single factor in the likelihood function is given by

$$\begin{aligned} & p(\boldsymbol{x}(t) \mid \boldsymbol{A}, r) \\ & = \int p(\boldsymbol{x}(t) \mid s(t), \boldsymbol{A}) r(s(t)) ds(t) \\ & = \int \prod_{j=1}^{n} \delta(x_j(t) \\ & - \sum_{i=1}^{n} A_{ji} s_i(t)) \prod_{i=1}^{n} r_i(s_i(t)) ds(t) \qquad (3) \\ & = |\det \mathbf{A}|^{-1} \prod_{i=1}^{n} r_i(\sum_{j=1}^{n} A_{ij}^{-1} x_j(t)), \qquad (4) \end{aligned}$$

where det denotes the determinat of a matrix. Then, we have

$$p(\boldsymbol{x}(t) \mid \boldsymbol{A}, r) = |\det \boldsymbol{A}|^{-1} r(\boldsymbol{A}^{-1} \boldsymbol{x}(t)). \qquad (5)$$

The log likelihood is

$$\log p(\boldsymbol{x}(t) | \boldsymbol{A}, r) = -\log |\det \boldsymbol{A}| + \log r(\boldsymbol{A}^{-1} \boldsymbol{x}(t)). \qquad (6)$$

Let us introduce $\boldsymbol{W} = \boldsymbol{A}^{-1}$ and $\boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t)$. With these quantities, the log likelihood (6) can be rewritten as

$$\log p(\boldsymbol{x}(t) | \boldsymbol{A}, r) = \log |\det \mathbf{W}| + \log r(\boldsymbol{y}(t)). \qquad (7)$$

The updating algorithm for learning the mixing matrix $\boldsymbol{A}$ in a linear generative model can be obtained from maximizing the log likelihood (6). Alternatively the recognition model $\boldsymbol{W}$ can be learned from maximizing (7). The detailed derivation of the neural learning algorithms for the maximization of (7) are discussed in Section 4.

Ⅲ. Relative Entropy Minimization

The goal of ICA is to extract independent components from their linear instantaneous mixtures. Let us denote the neural network (recognition model) output signals by $\boldsymbol{y}(t)$ and the neural network by $\boldsymbol{W}(t)$. For instance, in the linear feedforward neural network (see Figure 2), the output signals $\boldsymbol{y}(t)$ are given by

$$\boldsymbol{y}(t) = \boldsymbol{W}(t)\boldsymbol{x}(t), \qquad (8)$$

where $\boldsymbol{W}(t)$ is the synaptic weight matrix.

We briefly review the definitions of relative entropy and the mutual information to help the reader to understand the information-theoretic principle for ICA.

Definition 1 *The relative entropy or Kullback-Leibler distance between two probability density function* $p(x)$ *and* $q(x)$ *is defined as*

$$\begin{aligned} K[p(x)||q(x)] &= \int p(x) \log \frac{p(x)}{q(x)} dx \\ &= E\left\{ \log \frac{p(x)}{q(x)} \right\}, \end{aligned} \qquad (9)$$

where E denotes the statistical expectation. It is known [12] that the relative entropy is always nonnegative and is zero if and only if $p(x) = q(x)$.

Definition 2 *Consider two random variables* x *and* y *with a joint probability density function* $p(x, y)$ *and marginal probability density functions* $p(x)$ *and* $p(y)$. *The mutual information* $I(x, y)$ *is the relative entropy between the joint distribution and the product distribution* $p(x)p(y)$, *i.e.,*

$$\begin{aligned} I(x, y) &= \int \int p(x, y) \log \frac{p(x, y)}{p(x)p(y)} dxdy \\ &= K[p(x, y)||p(x)p(y)] \\ &= E\left\{ \log \frac{p(x, y)}{p(x)p(y)} \right\}. \end{aligned} \qquad (10)$$

For the minimization of statistical dependence, we choose the Kullback-Leibler distance which is an asymmetric measure between two different distributions. As an optimization, our risk function $R(\boldsymbol{W}(t))$ (which is the expectation of loss function $L(\boldsymbol{W}(t))$ is given by

$$\begin{aligned} R(\boldsymbol{W}(t)) &= E\{L(\boldsymbol{W}(t))\} \\ &= K\left[p(\boldsymbol{y}(t)) || \prod_{i=1}^{n} r_i(y_i(t)) \right] \\ &= \int p(\boldsymbol{y}(t)) \log \frac{p(\boldsymbol{y}(t))}{\prod_{i=1}^{n} r_i(y_i(t))} d\boldsymbol{y}(t) \\ &= E\{\log p(\boldsymbol{y}(t))\} - \sum_{i=1}^{n} E\{\log r_i(y_i(t))\}, \end{aligned} \qquad (11)$$

where $p(\boldsymbol{y}(t))$ is joint probability density of $\boldsymbol{y}(t)$ and $r_i(y_i(t))$ is the marginal probability density of $y_i(t)$. The Kullback-Leibler distance $K[p(\boldsymbol{y}(t)\|\prod_{i=1}^{n} r_i(y_i(t))]$ is nothing but mutual information $I(y_1(t),\cdots,y_n(t))$. It is known that when the risk function $R(\boldsymbol{W}(t))$ achieves the minimum, $\{y_i(t)\}$ are statistically independent.

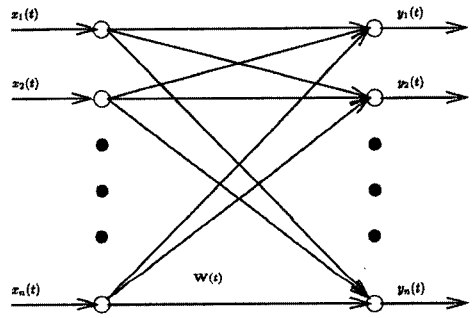

Figure 2: The structure of the feedforward network for the recognition model is shown. The observations $x_1(t),\cdots,x_n(t)$ are fed into the network, and the network output $y_1(t),\cdots,y_n(t)$ come out. The synaptic weight matrix $\boldsymbol{W}(t)$ is learned by the ICA algorithm (26).

Ⅳ. Learning Algorithms for Recognition Models

The recognition model to recover the independent source signals from their instantaneous mixtures, is implemented by a linear feedforward network or a linear feedback network. Using natural Riemannian gradient learning, ICA algorithms are rigorously derived for a linear feedback network and a linear feedback network.

Ⅳ-1. Natural Riemannian Gradient Learning

When the parameter space where a risk (or loss) function is defined, is a Euclidean space with an orthonormal coordinate system, the conventional gradient find the steepest descent direction. However, when the coordinate system is non-orthonormal, natural Riemannian gradient is able to find the steepest descent direction, whereas the conventional gradient could not. In this paper, the natural Riemannian gradient is adopted to derive efficient learning algorithms for ICA. The more details on natural Riemannian gradient learning can be found in Amari's recent work [1]. First we revisit ICA algorithm for a linear feedforward neural network using natural Riemannian gradient. Then we consider a linear feedback neural network and derive ICA algorithm rigorously.

Ⅳ-2. Linear Feedforward Network

Consider a linear feedforward neural network (see Figure 2) whose output $\boldsymbol{y}(t)$ is described by

$$\boldsymbol{y}(t) = \boldsymbol{W}(t)\boldsymbol{x}(t), \tag{12}$$

where $\boldsymbol{W}(t) \in \mathbb{R}^{n\times n}$ is a connection weight matrix. The (i,j)th element of the synaptic weight matrix $\boldsymbol{W}(t)$ is denoted by $W_{ij}(t)$ which represents the connection strength between the output signal $y_i(t)$ and the input signal $x_j(t)$. To extract independent components or to recover the source signals, the synaptic weight matrix $\boldsymbol{W}(t)$ needs to be updated such that in steady state, the components of $\boldsymbol{y}(t)$ are mutually independent.

The log likelihood function (7) or the risk function (11) require the knowledge of the probability densities of source signals, $\{r_i(\cdot)\}$ which are unknown to us. Thus, we use a hypothesized density model denoted by $q_i(\cdot)$. Using the hypothesized density, the risk function (11) is given by

$$\begin{aligned} R(\boldsymbol{W}(t)) &= L(\boldsymbol{W}(t)) \\ &= \int p(y)\log\frac{p(y)}{q_1(y_1)\cdots q_n(y_n)}dy \\ &= E\left\{\log\frac{p(y)}{q_1(y_1)\cdots q_n(y_n)}\right\} \\ &= -H(\boldsymbol{y}(t)) + \sum_{i=1}^{n} H(y_i(t)), \quad (13) \end{aligned}$$

where $H(\cdot)$ represent the (differential) entropy which are defined by

$$\begin{aligned} H(\boldsymbol{y}(t)) &= -\int \boldsymbol{y}(t)\log p(\boldsymbol{y}(t))d\boldsymbol{y}(t), \\ H(y_i(t)) &= -\int y_i(t)\log q_i(y_i(t))dy_i(t). \quad (14) \end{aligned}$$

It can be easily shown that

$$H(\boldsymbol{y}(t)) = H(\boldsymbol{x}(t)) + \log|\det \boldsymbol{W}(t)|. \quad (15)$$

Note that the minimization of the loss function (16) is identical to maximize the log likelihood (7). It is also shown [13] that the maximum likelihood approach and the information maximization principle lead to the identical risk function in ICA. This is also related to the factorial coding approach [14]. For the maximum information transfer from the input node to the output node, mutual information among the output nodes should be minimized. All these different approaches converge to the identical risk function (11) in ICA.

Note that $H(\boldsymbol{x}(t))$ does not depend on $W(t)$. By stochastic approximation, our loss function is

$$L(\boldsymbol{W}(t)) = -\sum_{i=1}^{n}\log q_i(y_i(t)) - \log|\det \boldsymbol{W}(t)|. \quad (16)$$

Let us define a function,

$$f_i(y_i(t)) = -\frac{d\log q_i(y_i(t))}{dy_i(t)} \quad (17)$$

Then,

$$\begin{aligned} d\left\{-\sum_{i=1}^{n}\log q_i(y_i(t))\right\} &= \sum_{i=1}^{n} f_i(y_i(t))dy_i(t) \quad (18) \\ &= f^T(\boldsymbol{y}(t))d\boldsymbol{y}(t), \quad (19) \end{aligned}$$

where $f(\boldsymbol{y}(t)) = [f_i(y_1(t)\cdots f_n(y_n(t))]^T$ and $d\boldsymbol{y}(t)$ is given in terms of $d\boldsymbol{W}(t)$ as

$$d\boldsymbol{y}(t) = d\boldsymbol{W}(t)\ \boldsymbol{W}^{-1}(t)\,\boldsymbol{y}(t). \quad (20)$$

Define a modified coefficient differential $d\boldsymbol{V}(t)$ as

$$d\boldsymbol{V}(t) = d\boldsymbol{W}(t)\ \boldsymbol{W}^{-1}(t). \quad (21)$$

With this definition, we have

$$d\left\{-\sum_{i=1}^{n}\log q_i(y_i(t))\right\} = f^T(\boldsymbol{y}(t))d\boldsymbol{V}(t)\,\boldsymbol{y}(t). \quad (22)$$

We calculate an infinitesimal increment of $\log|\det \boldsymbol{W}(t)|$, then we have

$$d\{\log|\det \boldsymbol{W}(t)|\} = \mathrm{Tr}\{d\boldsymbol{V}(t)\}, \quad (23)$$

where $\mathrm{Tr}\{\cdot\}$ denotes the trace which adds up all diagonal elements.

Thus combining (22) and (23) gives

$$dL(\boldsymbol{W}(t)) = f^T(\boldsymbol{y}(t))d\boldsymbol{V}(t)\,\boldsymbol{y}(t) - \mathrm{Tr}\{d\boldsymbol{V}(t)\}. \quad (24)$$

The differential in (24) is in terms of the modified coefficient differential matrix $d\boldsymbol{V}(t)$. Note that $\boldsymbol{V}(t)$ is a linear combination of the coefficient differentials $dW_{ij}(t)$. Thus, as long as $d\boldsymbol{W}(t)$ is nonsingular, $d\boldsymbol{V}(t)$ represents a valid search direction to minimize (16), because $d\boldsymbol{V}(t)$ spans the same tangent space of matrices as spanned by $d\boldsymbol{W}(t)$. This leads to a stochastic gradient learning algorithm for $\boldsymbol{V}(t)$ given by

$$\begin{aligned} \triangle\boldsymbol{V}(t) &= \boldsymbol{V}(t+1) - \boldsymbol{V}(t) \\ &= -\eta_t\frac{dL(\boldsymbol{W}(t))}{d\boldsymbol{V}(t)} \\ &= \eta_t\{\boldsymbol{I} - f(\boldsymbol{y}(t))\boldsymbol{y}^T(t)\}, \quad (25) \end{aligned}$$

where $\eta_t > 0$ is the learning rate. Thus the learning algorithm for updating $\boldsymbol{W}(t)$ is described by

$$\begin{aligned} \triangle\boldsymbol{W}(t) &= \boldsymbol{W}(t+1) - \boldsymbol{W}(t) \\ &= \eta_t\triangle\boldsymbol{V}(t)\,\boldsymbol{W}(t) \\ &= \eta_t\{\boldsymbol{I} - f(\boldsymbol{y}(t))\boldsymbol{y}^T(t)\}\boldsymbol{W}(t). \quad (26) \end{aligned}$$

Note that the conventional gradient $\frac{dL(\boldsymbol{W}(t))}{d\boldsymbol{W}(t)}$ postmultiplied by $\boldsymbol{W}^T(t)\,\boldsymbol{W}(t)$ is known as the natural gradient [9] in ICA. The algorithm (26) can also be derived from

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) - \eta_t\frac{dL(\boldsymbol{W}(t))}{d\boldsymbol{W}(t)}\ \boldsymbol{W}^T(t)\,\boldsymbol{W}(t). \quad (27)$$

Ⅳ-3. Linear Feedback Network

The linear feedback network for ICA has been used first by Jutten and Herault [3], although the algorithm was derived heuristically. In this paper, we consider a fully connected linear feedback network (see Figure 3) and derive an associated ICA algorithm rigorously using *natural Riemannian gradient*.

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

$$\begin{aligned}\boldsymbol{y}(t) &= \boldsymbol{x}(t)+\widehat{\boldsymbol{W}}(t)\boldsymbol{y}(t)\\ &= [\boldsymbol{I}-\widehat{\boldsymbol{W}}(t)]^{-1}\boldsymbol{x}(t).\end{aligned} \tag{28}$$

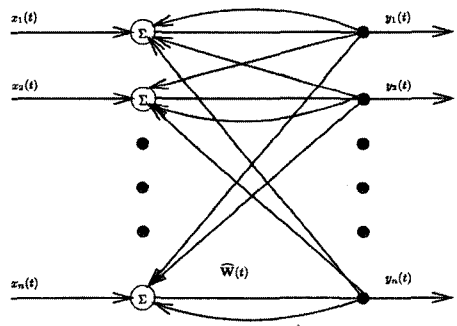

Figure 3: The structure of the feedback network for the recognition model is shown. The observations $x_1(t),\cdots,x_n(t)$ are fed into the network, and the network output $y_1(t),\cdots,y_n(t)$ are fed back to the summation node with connection strength $\widehat{\boldsymbol{W}}(t)$. The synaptic weight matrix $\widehat{\boldsymbol{W}}(t)$ is learned by the ICA algorithm (36).

Our loss function for the minimization of mutual information among output variables $y_i(t)$ is given by

$$\begin{aligned}L(\widehat{\boldsymbol{W}}(t)) &= -\sum_{i=1}^{n}\log q_i(y_i(t))\\ &\quad -\log\det|(\boldsymbol{I}-\widehat{\boldsymbol{W}}(t))^{-1}|.\end{aligned} \tag{29}$$

Using the definition of $f_i(y_i(t))$ in (17), we have

$$\begin{aligned}&d\Big\{-\sum_{i=1}^{n}\log q_i(y_i(t))\Big\}\\ &= f^T(\boldsymbol{y}(t))[\boldsymbol{I}-\widehat{\boldsymbol{W}}(t)]^{-1}d\widehat{\boldsymbol{W}}(t)\,\boldsymbol{y}(t).\end{aligned} \tag{30}$$

Define a modified coefficient differential $d\widehat{\boldsymbol{V}}(t)$ as

$$d\widehat{\boldsymbol{V}}(t) = [\boldsymbol{I}-\widehat{\boldsymbol{W}}(t)]^{-1}d\widehat{\boldsymbol{W}}(t). \tag{31}$$

With this definition, we have

$$d\Big\{-\sum_{i=1}^{n}\log q_i(y_i(t))\Big\} = f^T(\boldsymbol{y}(t))d\widehat{\boldsymbol{V}}(t)\,\boldsymbol{y}(t). \tag{32}$$

Similarly, it can be shown that

$$d\{\log\det|(\boldsymbol{I}-\widehat{\boldsymbol{W}}(t))^{-1}|\} = \mathrm{Tr}\{d\widehat{\boldsymbol{V}}(t)\}. \tag{33}$$

Thus combining (32) and (33) gives

$$dL(\widehat{\boldsymbol{W}}(t)) = f^T(\boldsymbol{y}(t))d\widehat{\boldsymbol{V}}(t)\,\boldsymbol{y}(t)-\mathrm{Tr}\{d\widehat{\boldsymbol{V}}(t)\}. \tag{34}$$

The differential in (34) is in terms of the modified coefficient differential $d\widehat{\boldsymbol{V}}(t)$. Since the differentials $d\,\widehat{V}_{ij}(t)$ are linear combinations of the basis $d\,\widehat{W}_{ij}(t)$, $d\widehat{\boldsymbol{V}}(t)$ represents a valid search direction to minimize (29). This leads to a stochastic gradient learning algorithm,

$$\begin{aligned}\triangle\widehat{\boldsymbol{V}}(t) &= \widehat{\boldsymbol{V}}(t+1)-\widehat{\boldsymbol{V}}(t)\\ &= -\eta(t)\frac{dL(\widehat{\boldsymbol{W}}(t))}{d\widehat{\boldsymbol{V}}(t)}\\ &= \eta t\{\boldsymbol{I}-f(\boldsymbol{y}(t))\boldsymbol{y}^T(t)\}.\end{aligned} \tag{35}$$

Thus, a learning algorithm for $\widehat{\boldsymbol{W}}(t)$ is given by

$$\begin{aligned}\triangle\widehat{\boldsymbol{W}}(t) &= \widehat{\boldsymbol{W}}(t+1)-\widehat{\boldsymbol{W}}(t)\\ &= [\boldsymbol{I}-\widehat{\boldsymbol{W}}(t)]\triangle\widehat{\boldsymbol{V}}(t)\\ &= \eta t[\boldsymbol{I}-\widehat{\boldsymbol{W}}(t)]\{\boldsymbol{I}-f(\boldsymbol{y}(t))\boldsymbol{y}^T(t)\}.\end{aligned} \tag{36}$$

Ⅳ-4. The Choice of Nonlinear Activation Function

From maximum likelihood and information-theoretic approaches, the optimal choice of a nonlinear function $f_i(y_i(t))$ is given by

$$f_i(y_i(t)) = -\frac{d\log r_i(y_i(t))}{dy_i(t)}. \tag{37}$$

Thus, if we have a priori knowledge of probability

density function of source signals, then we can select the optimal nonlinear function. However, in ICA, we do not have knowledge of probability density of source signals, so the hypothesized density $q_i(\cdot)$ was used in the derivation of the algorithm. Typical choice of nonlinear function $f_i(y_i(t))$ for sub-Gaussian signals (negative kurtosis), is cubic nonlinearity, i.e.; $f_i(y_i(t)) = y_i^3(t)$. Usually in digital communication application where transmitted signals are sub-Gaussian, cubic nonlinearity is successfully applied [15]. For super-Gaussian signals (positive kurtosis), $f_i(y_i(t)) = \tanh(\alpha y_i(t))$ works well. This nonlinear function can be obtained from $q_i(y_i(t)) \propto \mathrm{sech}(\alpha y_i(t))^{\frac{1}{\alpha}}$. This can be applied for separation of speech signals which are super-Gaussian. These two nonlinearities are exemplary choices.

V. Discussions

The loss function (16) can be extended to the case where we have more sensors than sources. The extended case is discussed here. In addition, the slight modification of the algorithms (26) and (36) with soft constraint is discussed.

V-1. The Extension to More Sensors than Sources

For the case where we have more sensors than sources, the loss function (16) can be modified as

$$L(\boldsymbol{W}(t)) = -\sum_{i=1}^{n} \log q_i(y_i(t)) - \frac{1}{2}\log\det \boldsymbol{W}^T(t)\,\boldsymbol{W}(t). \quad (38)$$

We calculate an infinitesimal increment of $\log\det \boldsymbol{W}^T(t)\,\boldsymbol{W}(t)$, then we have

$$d\{\log\det \boldsymbol{W}^T(t)\,\boldsymbol{W}(t)\} = \mathrm{Tr}\{[\boldsymbol{W}^T(t)\,\boldsymbol{W}(t)]^{-1} d\boldsymbol{W}(t)\,\boldsymbol{W}(t) + [\boldsymbol{W}^T(t)\,\boldsymbol{W}t]^{-1}\,\boldsymbol{W}^T(t)\,d\boldsymbol{W}(t)\}. \quad (39)$$

Thus we have

$$\frac{d}{d\boldsymbol{W}(t)}\{\log\det \boldsymbol{W}^T(t)\,\boldsymbol{W}(t)\} = 2\,\boldsymbol{W}(t)[\boldsymbol{W}^T(t)\,\boldsymbol{W}(t)]^{-1}. \quad (40)$$

Therefore, the learning algorithm for $\boldsymbol{W}(t)$ has the form of

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \eta(t)\{\boldsymbol{I} - f(\boldsymbol{y}(t))\boldsymbol{y}^T(t)\}\,\boldsymbol{W}(t). \quad (41)$$

This is identical to (26).

V-2. Soft Constraint

The learning algorithms (26) and (36) impose the restriction that all extracted signals $\{y_i(t)\}$ have

$$E\{f_i(y_i(t))\,y_i(t)\} = 1. \quad (42)$$

For nonstationary signals, this restriction might degrade the performance. Thus we introduce the soft constraint which eliminate the restriction (42). The resulting learning algorithm which is the modification of (26) is given by

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \eta t\{\boldsymbol{\Lambda}(t) - f(\boldsymbol{y}(t))y^T(t)\}\,\boldsymbol{W}(t), \quad (43)$$

where $\boldsymbol{\Lambda}(t)$ is the diagonal matrix whose ith diagonal element is given by $f_i(y_i(t))\,y_i(t)$. Similar modified algorithm for the linear feedback network can be also obtained.

VI. Computer Simulations

VI-1. Simulation 1

Three source signals were drawn from uniform distribution between -.5 and .5. The observation vector $\boldsymbol{x}(t)$ was generated via randomly generated linear

mapping $\boldsymbol{A}$,

$$\boldsymbol{x}(t) = \boldsymbol{A}\boldsymbol{s}(t), \tag{44}$$

where

$$\boldsymbol{A} = \begin{bmatrix} 0.1291 & 0.9505 & 0.1597 \\ 0.6048 & 0.3367 & 0.7808 \\ 0.5040 & 0.0924 & 0.6925 \end{bmatrix} \tag{45}$$

To recover the source signals, the linear feedforward network as shown in Figure 2 was tested. The cubic nonlinear function $f_i(y_i(t)) = y_i^3(t)$ was used. The learning rate was set to be constant, $\eta_t = .001$. After 10000 iterations, the synaptic weight matrix $\boldsymbol{W}(t)$ converges to

$$\boldsymbol{W}(10000)\boldsymbol{A} = \begin{bmatrix} 0.0073 & \mathbf{0.8671} & -0.0110 \\ \mathbf{0.8584} & -0.0042 & -0.0051 \\ -0.0019 & -0.0012 & \mathbf{0.8633} \end{bmatrix}. \tag{46}$$

It can be observed that the global transformation matrix $\boldsymbol{G} = \boldsymbol{WA}$ is close to the generalized permutation matrix which implies that the extracted signals $\boldsymbol{y}(t)$ are the scaled estimates of source signals $\boldsymbol{s}(t)$ up to certain permutation ambiguity. In addition, for quantative analysis, the performance index [15] which represents how close a global system matrix $\boldsymbol{G} = \boldsymbol{WA}$ is to a permutation matrix, is calculated. At steady state, the performance index was around between -30dB and -40dB.

VI-2. Simulation 2

The second simulation was conducted with speech signals. Three different original speech signals (which are unknown to us, see Figure 4) were transformed via a linear mapping given in (45). Three mixture signals are shown in Figure 5. For the recognition of speech signals, the linear feedforward neural network as shown in Figure 2 and the linear feedback network were used with associated learning algorithms (26) and (36). Using the nonlinear function $f_i(y_i(t)) = \tanh(10y_i(t))$, the synaptic weight matrix $\boldsymbol{W}(t)$ or $\widehat{\boldsymbol{W}}(t)$ were learned. The constant learning rate, $= \eta_t = .0003$ was used. After 4 epoch, the network output signals $\boldsymbol{y}(t)$ are shown in Figure 6 and 7. It can be observed that the network output signals $y_1(t), y_2(t), y_3(t)$ correspond to the original speech signals $s_2(t), s_3(t), s_1(t)$, respectively.

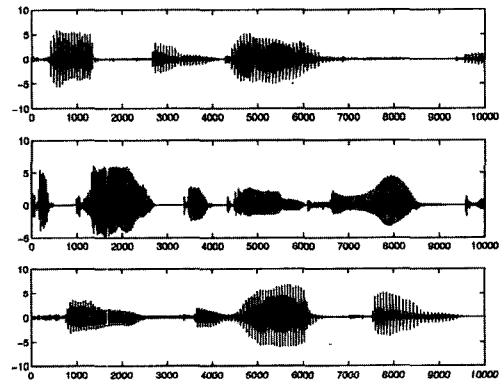

Figure 4: Original speech signals, $s_1(t), s_2(t), s_3(t)$ for the duration of 10000 samples.

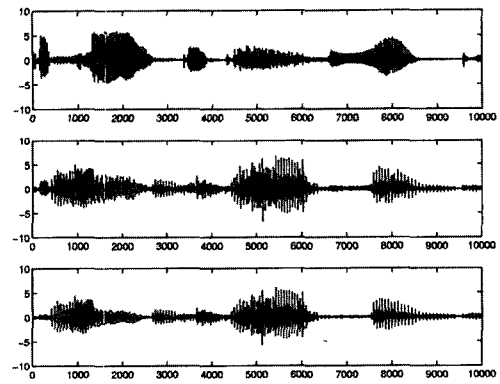

Figure 5: Mixture signals, $x_1(t), x_2(t), x_3(t)$ for the duration of 10000 samples.

Although the magnitude of network output signals are not identical to that of original speech signals, their

waveforms are well preserved. In addition, we define the signal to noise ratio improvement to see how the recovered signals are improved, compared to the mixtures. It is defined as

$$SNRI_{i=}\frac{E(x_i-s_i)^2}{E(y_i-s_i)^2}$$

The signal to noise ratio improvement for $i=1,2,3$ was 37dB, 22dB, 40dB.

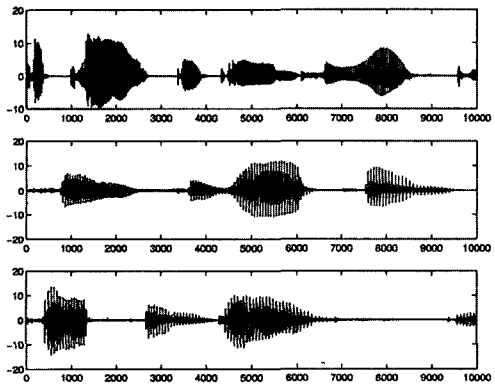

Figure 6: The recovered signals by the feedforward network with the algorithm (26), $y_1(t),y_2(t),y_3(t)$ for the duration of 10000 samples.

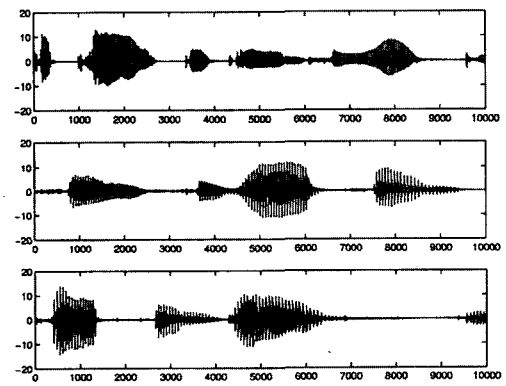

Figure 7: The recovered signals by the feedback network with the algorithm (36), $y_1(t),y_2(t),y_3(t)$ for the duration of 10000 samples.

Ⅶ. Conclusion

Maximum likelihood and the minimization of relative entropy approaches to ICA were presented and were shown to lead to the identical loss function for ICA. For recognition models, the linear feedforward network and the linear feedback network were presented. Associated learning algorithms for ICA using natural Riemannian gradient were derived rigorously. Theoretical results were confirmed by computer simulation results.

Acknowledgements

The author would like to thank Profs. S. Amari and A. Cichocki at Brain-style Information Systems Group, RIKEN in Japan for their helpful discussions.

References

[1] S. Amari, "Natural gradient works efficiently in learning," *Neural Computation*, vol. 10, pp. 251-276, 1998.

[2] P. Comon "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, pp. 287-314, 1994.

[3] C. Jutten and J. Herault, "Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture," *Signal Processing*, vol. 24, pp. 1-10, 1991.

[4] A. Cichocki, R. Unbehauen, L. Moszczynski, and E. Rummert, "A new on-line adaptive learning algorithm for blind separation of source signals," in *International Joint Conference on Neural Networks*, 1994, pp. 406-411.

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

[6] D. T. Pham, P Garrat, and C. Jutten, "Separation of mixture of independent sources through a maximum likelihood approach," in *European Signal Processing Conference*, 1992, pp. 771-774.

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

[8] J. F. Cardoso and B. H. Laheld, "Equivariant adaptive source separation," *IEEE Trans. Signal Processing*, vol. 44, pp. 3017-3030, 1996.

[9] 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.

[10] S. Choi, R. Liu, and A. Cichocki, "A spurious equlibria-free learning algorithm for the blind separation of non-zero skewness signals," *Neural Processing Letters*, vol. 7, pp. 1-8, 1998.

[11] S. Choi and A. Cichocki, "A linear feedforward neural network with lateral feedback connections," in *IEEE Signal Processing Workshop on Higher-order Statistics*, Banff, Canada, 1997, pp. 349-353.

[12] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, John Wiely & Sons, New York, 1991.

[13] J. F. Cardoso, "Infomax and maximum likelihood for source separation," *IEEE · Signal Processing Letters*, vol. 4, pp. 112-114, 1997.

[14] J. P. Nadal and N. Parga, "Nonlinear neurons in the low-noise limit: A factorial code maximize information transfer," *Network: Computation in Neural Systems*, vol. 5, pp. 565-581, 1994.

[15] S. Choi, "Adaptive blind signal separation for multiuser communications: An Information-theoretic approach," submitted for publication.

저 자 소 개

崔 丞 鎭 (正會員)

Seung-Jin Choi 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 University of Notre Dame, Indiana, in 1996. From August 1996 to January 1997, he was a Visiting Assistant Professor in the Department of Electrical Engineering at University of Notre Dame, Indiana. From February 1997 to August 1997, he was a frontier Researcher in the Laboratory for Artificial Brain Systems, RIKEN in Japan. In August 1997, he joined the School of Electrical and Electronics Engineering at Chungbuk National University where he is a Full-time Lecturer now. He also has been an Inbited Research Fellow at Brain-style Information Systems Research Group in Brain Science Institute, RIKEN in Japan. His current research interests include unsupervised learning algorithms, brain information processing, independent component analysis, and blind signal processing.