Differential Learning Algorithms for Decorrelation and Independent Component Analysis

Seungjin Choi

Department of Computer Science

Pohang University of Science and Technology

San 31 Hyoja-dong, Nam-gu

Pohang 790-784, Korea

Tel: +82-54-279-2259

Fax: +82-54-279-2299

Email: seungjin@postech.ac.kr

Revised on August 16, 2006

Accepted as a contributed article to

Neural Networks

Preferred section: Mathematical and Computational Analysis

Abstract

Decorrelation and its higher-order generalization, independent component analysis (ICA), are fundamental and important tasks in unsupervised learning, that were studied mainly in the domain of Hebbian learning. In this paper we present a variation of the natural gradient ICA, *differential ICA*, where the learning relies on the concurrent change of output variables. We interpret the differential learning as the maximum likelihood estimation of parameters with latent variables represented by the random walk model. In such a framework, we derive the differential ICA algorithm and, in addition, we also present the differential decorrelation algorithm that is treated as a special instance of the differential ICA. Algorithm derivation and local stability analysis are given with some numerical experimental results.

Keywords: Blind source separation, decorrelation, differential learning, Hebbian learning, independent component analysis.

1 Introduction

Hebb rules have been widely used in the domain of unsupervised learning and self-organization where no target value is available. It is a correlation learning that is based on the hypothesis of Hebb (Hebb, 1949) which states that the concurrent activation of neurons increases the strength of a connection between them. The Hebb rule was shown to be an output variance maximizer and its optimization properties were examined in detail (Linsker, 1988). It is also related with the maximum information preservation (which is known as *infomax*) between input and output variables, with assuming Gaussian characteristics implicitly. In contrast to the Hebb rule, the anti-Hebb rule (Földiák, 1989; Földiák, 1990) updates the synaptic weights in such a way that cross-correlations between associated nodes are minimized. Hence, it is an output variance minimizer and decorrelates associated output variables.

As an alternative to the conventional Hebbian learning, the differential Hebb rule was introduced in an ad-hoc manner and was examined in (Kosko, 1986). The main motivation of the differential Hebb rule is that concurrent change, rather than just concurrent activation, more accurately captures the *concomitant variation* that is central to inductively inferred functional relationships (Kosko, 1986). Under the assumption of Martingale processes, the differential Hebb rule was shown to be a covariance learning rather than a correlation learning. The differential anti-Hebb rule is a direct modification of the anti-Hebb rule. It updates the synaptic weights in a linear feedback network in such a way that the concurrent change of neurons is minimized. In this sense one can argue that the differential Hebb rule is an output differential variance minimizer. It was first introduced in (Choi, 1998) and its generalization with adopting a nonlinear function was applied to the problem of independent component analysis (ICA) (Choi, 1998).

In this paper we present a *differential ICA* algorithm, where the parameter learning relies on the concurrent change of output variables. To this end, we interpret the differential learning

as the maximum likelihood estimation of a linear generative model with assuming a random walk model for latent variables. In such a framework, we also employ the natural gradient method (Amari, 1998) that has been widely used for ICA and blind source separation (Amari et al., 1997a; Amari et al., 1997b; Choi et al., 1999; Choi et al., 2003), in order to derive our differential ICA algorithm. The differential ICA provides a simple way of taking some temporal structure of latent variables into account, through a random walk model. Although the differential ICA can be viewed as a slight variation of the natural gradient ICA (Amari et al., 1997a), our framework where the differential learning is interpreted as the maximum likelihood estimation with a random walk model, provides a principled way of interpreting the differential learning. In addition, we also present a natural gradient algorithm for differential decorrelation that can be treated as a special instance of the differential ICA.

The rest of this paper is organized as follows. Next section briefly reviews the natural gradient ICA algorithm. Sec. 3 provides our main contribution, illustrating the differential learning in the framework of the maximum likelihood estimation with a random walk model and presenting the derivation of the differential ICA algorithm using the natural gradient method. Sec. 4 presents a natural gradient algorithm for differential decorrelation that can be viewed a special instance of the differential ICA. Local stability analysis is carried out in Sec. 5. Several numerical examples are provided in Sec. 6, emphasizing the useful behavior of the differential ICA, with audio examples and face recognition task. Conclusions are drawn in Sec. 7.

2 Independent Component Analysis

Independent component analysis (ICA) is a statistical method, the goal of which is to learn non-orthogonal basis vectors from a set of observation data with basis coefficients being statistically independent. A simple noise-free linear generative model assumes that the observed data $\boldsymbol{x}(t) \in$

$\mathbb{R}^n$ is generated by a linear combination of basis vectors $\boldsymbol{a}_i \in \mathbb{R}^n$, i.e.,

$$\begin{aligned} \boldsymbol{x}(t) &= \sum_{i=1}^{n} \boldsymbol{a}_i s_i(t) \\ &= \boldsymbol{A}\boldsymbol{s}(t), \end{aligned} \tag{1}$$

where $\boldsymbol{s}(t) = [s_1(t) \cdots s_n(t)]$ is an n-dimensional vector containing basis coefficients $\{s_i(t)\}$ and $\boldsymbol{A} = [\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n] \in \mathbb{R}^{n \times n}$ is a collection of basis vectors in its columns, which is known as a mixing matrix in source separation. Given a set of observation data $\{\boldsymbol{x}(t)\}$, ICA learns basis vectors $\boldsymbol{a}_i$ in such a way that statistical dependence among basis coefficients $\{s_i(t)\}$ is minimized. Hence $\{s_i(t)\}$ are called independent components or sources.

It is known that ICA performs source separation, the goal of which is to restore unknown sources without resorting to any prior knowledge, given only a set of observation data. Source separation is achieved by estimating the mixing matrix $\boldsymbol{A}$ or its inverse $\boldsymbol{W} = \boldsymbol{A}^{-1}$ (which is known as the *demixing matrix*).

A single factor of the likelihood function leads to the following objective function (MacKay, 1996; Amari et al., 1997a)

$$\mathcal{J}_1 = -\log|\det \boldsymbol{W}| - \sum_{i=1}^{n} \log p_i(y_i(t)), \tag{2}$$

where $\boldsymbol{y}(t) = [y_1(t), \ldots, y_n(t)]^\top = \boldsymbol{W}\boldsymbol{x}(t)$ and $p_i(\cdot)$ represents the hypothesized probability density function for the latent variable $s_i(t)$ (or its estimate $y_i(t)$). The objective function (2) can also be derived in the framework of mutual information minimization (Lee et al., 2000).

The well-known natural gradient ICA algorithm which iteratively finds a minimum of (2), has the form

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \eta_t \left\{ \boldsymbol{I} - \varphi(\boldsymbol{y}(t))\boldsymbol{y}^\top(t) \right\} \boldsymbol{W}(t), \tag{3}$$

where $\eta_t > 0$ is a learning rate and $\varphi(\boldsymbol{y}) = [\varphi_1(y_1), \ldots, \varphi_n(y_n)]^\top$ is an n-dimensional vector, each element of which corresponds to the negative score function, i.e., $\varphi_i(y_i) = -\frac{d \log p_i(y_i)}{dy_i}$

where $p_i(\cdot)$ is the hypothesized probability density function for s_i. More details on ICA or source separation can be found in (Hyvärinen et al., 2001; Cichocki & Amari, 2002; Choi et al., 2005) (and references therein).

3 Differential ICA

In a wide sense, most of ICA algorithms based on unsupervised learning belong to the Hebb-type rule or its generalization with adopting nonlinear functions. Motivated from the differential Hebb rule (Kosko, 1986) and differential decorrelation (Choi, 2002; Choi, 2003), we introduce an ICA algorithm employing the differential learning and natural gradient, which leads to a differential ICA algorithm. We first introduce a random walk model for latent variables, in order to show that the differential learning is interpreted as the maximum likelihood estimation of a linear generative model. Then the detailed derivation of the differential ICA algorithm is presented.

3.1 Random Walk Model for Latent Variables

Given a set of observation data, $\{\boldsymbol{x}(t)\}$, the task of learning the linear generative model (1) under a constraint of latent variables being statistically independent, is a semiparametric estimation problem. The maximum likelihood estimation of basis vectors $\{\boldsymbol{a}_i\}$ involves a probabilistic model for latent variables which are treated as nuisance parameters.

In order to show a link between the differential learning and maximum likelihood estimation, we consider a random walk model for latent variables $s_i(t)$, which is a simple Markov chain, i.e.,

$$s_i(t) = s_i(t-1) + \epsilon_i(t), \tag{4}$$

where the innovation $\epsilon_i(t)$ is assumed to have zero mean with a density function $q_i(\epsilon_i(t))$. In

addition, innovation sequences $\{\epsilon_i(t)\}$ are assumed to be mutually independent white sequences, i.e., they are spatially independent and temporally white as well.

Let us consider latent variables $s_i(t)$ over an N-point time block. We define the vector $\vec{\boldsymbol{s}}_i$ as

$$\vec{\boldsymbol{s}}_i = [s_i(0), \ldots, s_i(N-1)]^\top. \tag{5}$$

Then the joint probability density function of $\vec{\boldsymbol{s}}_i$ can be written as

$$\begin{aligned} p_i(\vec{\boldsymbol{s}}_i) &= p_i\left(s_i(0), \ldots, s_i(N-1)\right) \\ &= \prod_{t=0}^{N-1} p_i(s_i(t)|s_i(t-1)), \end{aligned} \tag{6}$$

where $s_i(t) = 0$ for $t < 0$ and the statistical independence of innovation sequences was taken into account.

It follows from the random walk model (4) that the conditional probability density of $s_i(t)$ given its past samples can be written as

$$p_i(s_i(t)|s_i(t-1)) = q_i(\epsilon_i(t)). \tag{7}$$

Combining (6) and (7) leads to

$$\begin{aligned} p_i(\vec{\boldsymbol{s}}_i) &= \prod_{t=0}^{N-1} q_i(\epsilon_i(t)) \\ &= \prod_{t=0}^{N-1} q_i\left(s_i'(t)\right), \end{aligned} \tag{8}$$

where $s_i'(t) = s_i(t) - s_i(t-1)$ which is the first-order approximation of the differentiation.

Take the statistical independence of latent variables and (8) into account, then we can write the joint density $p(\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_n)$ as

$$\begin{aligned} p(\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_n) &= \prod_{i=1}^{n} p_i(\vec{\boldsymbol{s}}_i) \\ &= \prod_{t=0}^{N-1} \prod_{i=1}^{n} q_i\left(s_i'(t)\right). \end{aligned} \tag{9}$$

The factorial model given in (9) will be used as an optimization criterion in deriving the differential ICA algorithm as well as the differential decorrelation, which are described in Sec. 3.2 and Sec. 4.1, respectively in detail.

3.2 Algorithm

Denote a set of observation data by

$$\mathcal{X} = \{\vec{\boldsymbol{x}}_1, \ldots, \vec{\boldsymbol{x}}_n\}, \tag{10}$$

where

$$\vec{\boldsymbol{x}}_i = [x_i(0), \ldots, x_i(N-1)]^\top. \tag{11}$$

Then the normalized log-likelihood is given by

$$\begin{aligned} \frac{1}{N} \log p(\mathcal{X}|\boldsymbol{A}) &= -\log|\det \boldsymbol{A}| + \frac{1}{N} \log p(\vec{\boldsymbol{s}}_1, \ldots, \vec{\boldsymbol{s}}_n) \\ &= -\log|\det \boldsymbol{A}| + \frac{1}{N} \sum_{t=0}^{N-1} \sum_{i=1}^{n} \log q_i(s_i'(t)). \end{aligned} \tag{12}$$

Let us denote the inverse of $\boldsymbol{A}$ by $\boldsymbol{W} = \boldsymbol{A}^{-1}$. The estimate of latent variables is denoted by $\boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t)$. With these defined variables, the objective function (that is the negative normalized log-likelihood) is given by

$$\begin{aligned} \mathcal{J}_2 &= -\frac{1}{N} \log p(\mathcal{X}|\boldsymbol{A}) \\ &= -\log|\det \boldsymbol{W}| - \frac{1}{N} \sum_{t=0}^{N-1} \sum_{i=1}^{n} \log q_i(y_i'(t)), \end{aligned} \tag{13}$$

where s_i is replaced by its estimate y_i and $y_i'(t) = y_i(t) - y_i(t-1)$ (the first-order approximation of the differentiation).

For on-line learning, the sample average is replaced by the instantaneous value. Hence the on-line version of the objective function (13) is given by

$$\mathcal{J}_3 = -\log|\det \boldsymbol{W}| - \sum_{i=1}^{n} \log q_i(y_i'(t)), \tag{14}$$

Note that objective function (14) is slightly different from (2) used in the conventional ICA based on the minimization of mutual information or the maximum likelihood estimation.

We derive a natural gradient learning algorithm which finds a minimum of (14). To this end, we follow the way that was discussed in (Amari et al., 1997a; Amari, 1998; Choi et al., 2000). We calculate the total differential $d\mathcal{J}_3(\boldsymbol{W})$ due to the change $d\boldsymbol{W}$

$$\begin{aligned} d\mathcal{J}_3 &= \mathcal{J}_3(\boldsymbol{W}+d\boldsymbol{W}) - \mathcal{J}_3(\boldsymbol{W}) \\ &= d\left\{-\log|\det \boldsymbol{W}|\right\} + d\left\{-\sum_{i=1}^{n}\log q_i(y_i'(t))\right\}. \end{aligned} \tag{15}$$

Define

$$\varphi_i(y_i') = -\frac{d\log q_i(y_i')}{dy_i'}. \tag{16}$$

and construct a vector $\varphi(\boldsymbol{y}') = [\varphi_1(y_1'), \ldots, \varphi_n(y_n')]^\top$.

With this definition, we have

$$\begin{aligned} d\left\{-\sum_{i=1}^{n}\log q_i(y_i'(t))\right\} &= \sum_{i=1}^{n}\varphi_i(y_i'(t))dy_i'(t) \\ &= \varphi^\top(\boldsymbol{y}'(t))d\boldsymbol{y}'(t). \end{aligned} \tag{17}$$

One can easily see that

$$d\left\{-\log|\det \boldsymbol{W}|\right\} = \mathrm{tr}\left\{d\boldsymbol{W}\boldsymbol{W}^{-1}\right\}. \tag{18}$$

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

$$d\boldsymbol{V} = d\boldsymbol{W}\boldsymbol{W}^{-1}. \tag{19}$$

Then, with this modified differential matrix, the total differential $d\mathcal{J}_3(\boldsymbol{W})$ is computed as

$$d\mathcal{J}_3 = -\mathrm{tr}\left\{d\boldsymbol{V}\right\} + \varphi^\top(\boldsymbol{y}'(t))d\boldsymbol{V}\boldsymbol{y}'(t). \tag{20}$$

A gradient descent learning algorithm for updating $\boldsymbol{V}$ is given by

$$\begin{aligned} \boldsymbol{V}(t+1) &= \boldsymbol{V}(t) - \eta_t \frac{d\mathcal{J}_3}{d\boldsymbol{V}} \\ &= \eta_t \left\{ \boldsymbol{I} - \varphi(\boldsymbol{y}'(t)) \boldsymbol{y}'^{\top}(t) \right\}. \end{aligned} \tag{21}$$

Hence, it follows from the relation (19) that the updating rule for $\boldsymbol{W}$ has the form

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \eta_t \left\{ \boldsymbol{I} - \varphi(\boldsymbol{y}'(t)) \boldsymbol{y}'^{\top}(t) \right\} \boldsymbol{W}(t). \tag{22}$$

Remarks

- The algorithm (22) was originally derived in an *ad hoc* manner in (Choi, 1998). Here we show that the algorithm (22) can be derived in the framework of maximum likelihood estimation and a random walk model.

- The algorithm (22) can be viewed as a special case of the temporal ICA algorithm (Attias & Schreiner, 1998) where the spatiotemporal generative model was employed.

- In the conventional ICA algorithm, the nonlinear function $\varphi_i(\cdot)$ depends on the probability distribution of source. However, in the differential ICA algorithm, the nonlinear function is chosen, depending on the probability distribution of $\epsilon_i(t) = s_i(t) - s_i(t-1)$, i.e., the difference of adjacent latent variables in the time domain. In general, the innovation is more non-Gaussian, compared to the signal itself. In this sense, the differential ICA algorithm works better than the conventional ICA algorithm when source was generated by a linear combination of innovation and its time-delayed replica (e.g., moving average). This is confirmed by a simple numerical example.

- As in the flexible ICA (Choi et al., 2000), we can adopt a flexible nonlinear function based on the generalized Gaussian distribution.

4 Differential Decorrelation

4.1 Algorithm

In general, decorrelation is treated as a variance minimizer, which implicitly assumes Gaussian latent variables in the framework of maximum likelihood learning a probabilistic linear generative model. Here we derive a differential decorrelation algorithm using the factorial model (9) with assuming that $y_i'(t)$ is Gaussian. In such a special case, the objective function (14) is simplified as

$$\mathcal{J}_4 = -\log|\det \boldsymbol{W}| + \frac{1}{2}\sum_{i=1}^{n}\frac{[y_i'(t)]^2}{\lambda_i(t)}, \tag{23}$$

where $\lambda_i(t)$ is the differential variance that is estimated by

$$\lambda_i(t) = (1-\delta)\lambda_i(t-1) + \delta y_i'^2(t), \tag{24}$$

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

We employ a natural gradient learning method to derive an updating rule to find a minimum of the objective function (23). The derivation is carried out in a similar manner that was used in Sec. 3.2.

We calculate the total differential $d\mathcal{J}_4(\boldsymbol{W})$ due to the change $d\boldsymbol{W}$

$$\begin{aligned} d\mathcal{J}_4(\boldsymbol{W}) &= \frac{1}{2}d\left\{\sum_{i=1}^{n}\frac{[y_i'(t)]^2}{\lambda_i(t)}\right\} - d\left\{\log|\det \boldsymbol{W}|\right\} \\ &= \sum_{i=1}^{n}\frac{y_i'(t)dy_i'(t)}{\lambda_i(t)} - \mathrm{tr}\left\{d\boldsymbol{W}\boldsymbol{W}^{-1}\right\} \\ &= \boldsymbol{y}'^{\top}(t)\boldsymbol{\Lambda}^{-1}(t)d\boldsymbol{V}\boldsymbol{y}'(t) - \mathrm{tr}\left\{d\boldsymbol{V}\right\}, \end{aligned} \tag{25}$$

where $\boldsymbol{\Lambda}(t)$ is a diagonal matrix containing the differential variance $\lambda_i(t)$ in its diagonal entries. The nonholonomic basis $d\boldsymbol{V}$ is defined by $d\boldsymbol{V} = d\boldsymbol{W}\boldsymbol{W}^{-1}$ as in (19). Thus, a natural gradient learning algorithm which finds a minimum solution to the objective function (23), has the form

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \eta_t\left\{\boldsymbol{I} - \boldsymbol{\Lambda}^{-1}(t)\boldsymbol{y}'(t)\boldsymbol{y}'^{\top}(t)\right\}\boldsymbol{W}(t). \tag{26}$$

4.2 Alterative View of Differential Decorrelation

We provides an alternative view of the differential decorrelation algorithm (26). To this end, we consider the following objective function:

$$\mathcal{J}_5(\boldsymbol{W}) = \frac{1}{2}\left\{\sum_{i=1}^{n}\log E\{y_i'^2(t)\} - \log\det\left(E\left\{\boldsymbol{y}'(t)\boldsymbol{y}'^{\top}(t)\right\}\right)\right\}, \tag{27}$$

where $\boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t)$. The objective function (27) is a non-negative function which takes minima if and only if $E\{y_i'(t)y_j'(t)\} = 0$, for $i, j = 1, \ldots, n,\ i \neq j$. The objective function (27) a direct consequence of the Hadamard's inequality. In fact the objective function (27) is a slight modification of the one in (Matsuoka et al., 1995), replacing output values by their differentiated values. The derivation of the natural gradient algorithm which finds a minimum of (27) is given below.

We calculate the total differential $d\mathcal{J}_5(\boldsymbol{W})$ due to the change $d\boldsymbol{W}$

$$\begin{aligned} d\mathcal{J}_5(\boldsymbol{W}) &= \mathcal{J}_5(\boldsymbol{W}+d\boldsymbol{W}) - \mathcal{J}_5(\boldsymbol{W}) \\ &= \frac{1}{2}d\left\{\sum_{i=1}^{n}\log E\{y_i'^2(t)\}\right\} - \frac{1}{2}d\left\{\log\det\left(E\{\boldsymbol{y}'(t)\boldsymbol{y}'^{\top}(t)\}\right)\right\} \\ &= \sum_{i=1}^{n}\frac{E\{y_i'(t)dy_i'(t)\}}{E\{y_i'^2(t)\}} - \mathrm{tr}\{(\boldsymbol{W}^{-1}d\boldsymbol{W}\} - \frac{1}{2}d\left\{\log\det\boldsymbol{C}_{x'x'}(t)\right\}, \end{aligned} \tag{28}$$

where $\boldsymbol{C}_{x'x'}(t)$ is the differential correlation matrix of $\boldsymbol{x}(t)$ defined by

$$\boldsymbol{C}_{x'x'}(t) = E\left\{\boldsymbol{x}'(t)\boldsymbol{x}'^{\top}(t)\right\}. \tag{29}$$

Define a modified differential matrix $d\boldsymbol{V} = d\boldsymbol{W}\boldsymbol{W}^{-1}$ as in (19). Recall that the differential variance matrix was denoted by $\boldsymbol{\Lambda}(t)$ whose diagonal entries are estimated by (24).

With these defined matrices, the total differential $d\mathcal{J}_5(\boldsymbol{W})$ can be written as

$$d\mathcal{J}_5(\boldsymbol{W}) = E\{\boldsymbol{y}'^{\top}(t)\boldsymbol{\Lambda}^{-1}(t)d\boldsymbol{V}\boldsymbol{y}'(t)\} - \mathrm{tr}\{d\boldsymbol{V}\} - \frac{1}{2}d\left\{\log\det\boldsymbol{C}_{x'x'}(t)\right\}. \tag{30}$$

Hence, the gradient of the objective function (27) with respect to the modified differential matrix $d\boldsymbol{V}$ is given by

$$\frac{d\mathcal{J}_5(\boldsymbol{W})}{d\boldsymbol{V}} = E\left\{\boldsymbol{\Lambda}^{-1}(t)\boldsymbol{y}'(t)\boldsymbol{y}'^{\top}(t)\right\} - \boldsymbol{I}. \tag{31}$$

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

$$\boldsymbol{V}(t+1) = \boldsymbol{V}(t) + \eta_t\left\{\boldsymbol{I} - \boldsymbol{\Lambda}^{-1}(t)\boldsymbol{y}'(t)\boldsymbol{y}'^{\top}(t)\right\}, \tag{32}$$

where $\eta_t > 0$ is the learning rate. It follows from the definition (19) that the learning algorithm for $\boldsymbol{W}$ is given by

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \eta_t\left\{\boldsymbol{I} - \boldsymbol{\Lambda}^{-1}(t)\boldsymbol{y}'(t)\boldsymbol{y}'^{\top}(t)\right\}\boldsymbol{W}(t), \tag{33}$$

which is identical to (26) The algorithm (33) is a differential version of the equivariant nonstationary source separation algorithm in (Choi et al., 2002).

Remarks

- The learning algorithm (33) can also be written as

$$\boldsymbol{W}(t+1) = \boldsymbol{W}(t) + \eta_t\boldsymbol{\Lambda}^{-1}(t)\left\{\boldsymbol{\Lambda}(t) - \boldsymbol{y}'(t)\boldsymbol{y}'^{\top}(t)\right\}\boldsymbol{W}(t). \tag{34}$$

  Thus this differential decorrelation algorithm has properties inherited from the nonholonomic ICA algorithms (Amari et al., 2000).

- The on-line version of $d\mathcal{J}_5(\boldsymbol{W})$ is identical to the differential $d\mathcal{J}_4(\boldsymbol{W})$, neglecting the term $d\left\{\log\det\boldsymbol{C}_{x'x'}(t)\right\}$ in (30) since it does not depend on $\boldsymbol{W}$. Objective functions (23) and (27) were motivated from different principles, however, they led to the same learning algorithm.

5 Local Stability Analysis

5.1 Analysis of Differential Decorrelation

In this section, we show that the stationary points of (26) are locally stable. To this end we calculate the Hessian $d^2\mathcal{J}_4$ in terms of the modified differential matrix $d\boldsymbol{V}$ and show that it is positive.

For shorthand notation, we omit the time index t in the following analysis. The Hessian $d^2\mathcal{J}_4$ is computed as

$$\begin{aligned} d^2\mathcal{J}_4 &= E\left\{\boldsymbol{y}'^{\top} d\boldsymbol{V}^{\top}\boldsymbol{\Lambda}^{-1} d\boldsymbol{V}\boldsymbol{y}' + \boldsymbol{y}'^{\top}\boldsymbol{\Lambda}^{-1} d\boldsymbol{V} d\boldsymbol{V}\boldsymbol{y}'\right\} \\ &= E\left\{\boldsymbol{y}'^{\top} d\boldsymbol{V}^{\top}\boldsymbol{\Lambda}^{-1} d\boldsymbol{y}'\right\} + E\left\{\boldsymbol{y}'^{\top}\boldsymbol{\Lambda}^{-1} d\boldsymbol{V} d\boldsymbol{y}'\right\} \\ &= \sum_{i,j}\frac{\lambda_i}{\lambda_j}(dv_{ji})^2 + \sum_{i,j} dv_{ij}dv_{ji}, \end{aligned} \tag{35}$$

where the statistical expectation is taken at the solution which satisfies the condition $E\{y'_i y'_j\} = 0$ for $i \neq j$.

For a pair (i,j), $i \neq j$, the summand in the first term in (35) can be rewritten as

$$\begin{aligned} &\frac{\lambda_i}{\lambda_j}(dv_{ji})^2 + \frac{\lambda_j}{\lambda_i}(dv_{ij})^2 + 2dv_{ij}dv_{ji} \\ &= \begin{bmatrix} dv_{ij} & dv_{ji} \end{bmatrix} \begin{bmatrix} \frac{\lambda_j}{\lambda_i} & 1 \\ 1 & \frac{\lambda_i}{\lambda_j} \end{bmatrix} \begin{bmatrix} dv_{ij} \\ dv_{ji} \end{bmatrix}, \end{aligned} \tag{36}$$

which is always non-negative. Hence $d^2\mathcal{J}_4$ is always positive. Therefore the algorithm (26) is locally stable around the solutions.

5.2 Analysis of Differential ICA

The differential ICA algorithm (22) can be obtained by replacing $\boldsymbol{y}(t)$ by $\boldsymbol{y}'(t)$ in the conventional ICA algorithm (3). Thus the local stability analysis of the algorithm (22) can be done in a similar

manner, following the result in (Amari et al., 1997a). As in (Amari et al., 1997a), we calculate the expected Hessian $E\{d^2\mathcal{J}_3\}$ (in which the expectation is taken at $\boldsymbol{W} = \boldsymbol{A}^{-1}$) in terms of the modified differential matrix $d\boldsymbol{V}$. For shorthand notation, we omit the time index t in the following analysis.

The expected Hessian $E\{d^2\mathcal{J}_3\}$ is given by

$$\begin{aligned} E\{d^2\mathcal{J}_3\} &= E\left\{\boldsymbol{y}' d\boldsymbol{V}^\top \boldsymbol{\Phi} d\boldsymbol{y}' + \varphi^\top(\boldsymbol{y}') d\boldsymbol{V}\, d\boldsymbol{y}'\right\} \\ &= E\left\{\boldsymbol{y}' d\boldsymbol{V}^\top \boldsymbol{\Phi} d\boldsymbol{V}\boldsymbol{y}' + \varphi^\top(\boldsymbol{y}') d\boldsymbol{V}\, d\boldsymbol{V}\boldsymbol{y}'\right\} \\ &= \sum_{j\neq i}\left[\sigma_i^2\kappa_j\,(dv_{ji})^2 + dv_{ij}dv_{ji}\right] + \sum_i(\zeta_i+1)\,(dv_{ii})^2\,, \end{aligned} \tag{37}$$

where the statistical expectation is taken at the solution so that $\{y_i\}$ are mutually independent and

$$\boldsymbol{\Phi} = \begin{bmatrix} \dot\varphi_1(y_1') & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \dot\varphi_n(y_n') \end{bmatrix} \tag{38}$$

$$\dot\varphi_i(y_i') = \frac{d\varphi_i(y_i')}{dy_i'} \tag{39}$$

$$\sigma_i^2 = E\{y_i'^2\} \tag{40}$$

$$\kappa_i = E\{\dot\varphi_i(y_i')\} \tag{41}$$

$$\zeta_i = E\{y_i'^2\dot\varphi_i(y_i')\}. \tag{42}$$

It follows from (37) that $E\{d^2\mathcal{J}_3\}$ is positive if and only if

$$\kappa_i > 0 \tag{43}$$

$$\zeta_i + 1 > 0 \tag{44}$$

$$\sigma_i^2\sigma_j^2\kappa_i\kappa_j > 1. \tag{45}$$

6 Numerical Examples

6.1 Example 1

A simple numerical example is given to evaluate the validity of the differential decorrelation algorithm (26). Three independent colored Gaussian random variables is linearly mixed to generate the observation vector $\boldsymbol{x}(t)$ with a differential correlation matrix

$$\boldsymbol{C}_{x'x'} = \begin{bmatrix} 8.367 & 3.274 & 2.448 \\ 3.274 & 1.349 & 0.943 \\ 2.448 & 0.943 & 0.790 \end{bmatrix}. \tag{46}$$

We applied the algorithm (26) with the differential variance matrix being fixed as $\boldsymbol{\Sigma} = \boldsymbol{I}$ and with a constant learning rate, $\eta_t = .001$. Fig. 1 shows the evolution of $E\{y_1'(t)y_2'(t)\}$ as an example. Other differential correlations were also suppressed in a similar fashion.

6.2 Example 2

We present a simple numerical example to show the usefulness of our differential ICA algorithm which is described in (22). Three independent innovation sequences were drawn from Laplacian distribution. Each innovation sequence was convolved with a moving average filter (with exponentially decreasing impulse response) in order to generate colored sources. These sources were linearly mixed via 3×3 mixing matrix $\boldsymbol{A}$

We compare the performance of our differential ICA algorithm with that of the conventional natural gradient ICA algorithm in terms of the performance index (PI) which is defined as

$$\text{PI} = \frac{1}{2(n-1)} \sum_{i=1}^{n} \left\{ \left(\sum_{k=1}^{n} \frac{|g_{ik}|^2}{\max_j |g_{ij}|^2} - 1 \right) + \left(\sum_{k=1}^{n} \frac{|g_{ki}|^2}{\max_j |g_{ji}|^2} - 1 \right) \right\}, \tag{47}$$

where g_{ij} is the (i,j)-element of the global system matrix $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{A}$ and $\max_j g_{ij}$ represents the maximum value among the elements in the ith row vector of $\boldsymbol{G}$, $\max_j g_{ji}$ does the maximum

value among the elements in the ith column vector of $\boldsymbol{G}$. The performance index defined in (47) tells us how far the global system matrix $\boldsymbol{G}$ is from a generalized permutation matrix.

It is expected that the conventional ICA algorithm would have difficulty in separating these sources because they are close to Gaussian. The differential ICA algorithm inherently resort to the innovation sequence rather than the source itself (since it is motivated by a simple Markov model). The result of a numerical example is shown in Fig. 2. Hinton diagrams for the global matrix $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{A}$ for each method are shown in Fig. 3 where each square's area represents the magnitude of the element of the matrix and each square's color represents the sign of the element (dark for negative value and white for positive value). For successful separation, each row and column has only one dominant square (regardless of its color).

6.3 Example 3

A music audio signal was linearly mixed with a white Gaussian signal (see Fig. 4 for two original sources and Fig. 5 for their linear instantaneous mixtures). The music source signal was close to Gaussian. Since one source is white Gaussian, some existing source separation algorithms that exploit the temporal non-whiteness of sources, are not applicable to this case. The recovered signals by ICA and differential ICA, are shown in Figs. 6 and 7, respectively. One can see that the differential ICA restored original music signal and a white Gaussian noise source, very well, whereas the conventional ICA had difficulty. Hinton diagrams for the global matrix $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{A}$ for each method are shown in Fig. 8 where one can clearly see the high performance of the differential ICA, compared to the conventional ICA.

6.4 Example 4

An eigenface-based method (Sirovich & Kirby, 1987; Turk & Pentland, 1991) is a widely-used face recognition technique, where PCA is applied to an ensemble of vectors, each of which is constructed from a face image through row-by-row or column-by-column scanning. Factorial faces, computed by ICA, were also shown to be useful in face recognition (Bartlett et al., 1998; Choi & Lee, 2000). In this example, we used ORL face DB (Samaria & Harter, 1994) (40 people and 10 images for each person, i.e., $40 \times 10 = 400$ images in total) and applied PCA, ICA, and differential ICA for the task of face recognition. The output variables $y_i(t)$ computed by PCA, ICA, and differential ICA serve as features which are fed into a classifier. As an classifier, we used a simple nearest-neighbor classifier. Basis face images, computed by ICA and differential ICA, are shown in Fig. 9. Basis images computed by the standard ICA contains more visible structure of whole face of each person, whereas the basis images computed by the differential ICA contains better-visible face components or features of faces. In addition, in case of the standard ICA, face components (or features) for recognition are biased in several specific basis images. In contrast, in case of the differential ICA, face components (such as eye, noise, lip) are better-visible in basis images and are better-distributed through basis images, which are expected to give higher classification performance.

For each method, 5-fold cross-validation was applied, where each fold contains 80 face images. The recognition performance is summarized in Table 1, where the higher classification performance of the differential ICA is shown, compared to the standard ICA and PCA.

7 Discussion

In this paper we have introduced a method of differential learning for decorrelation and ICA and have presented natural gradient algorithms for differential decorrelation and differential ICA. We have interpreted the differential learning as a maximum likelihood estimation of a noise-free linear generative model with assuming a random walk model for latent variables. This gave a new insight to the differential learning. In this framework, we have presented natural gradient learning algorithms for differential ICA and decorrelation, and local stability analysis as well. Numerical examples such as an audio example and face recognition, have verified the useful behavior of the differential ICA, compared to the conventional ICA.

8 Acknowledgments

This work was supported by Korea Ministry of Commerce, Industry, and Energy under Brain Neuroinformatics Program and Korea MIC under ITRC support program supervised by the IITA (IITA-2005-C1090-0501-0018).

References

Amari, S. (1998). Natural gradient works efficiently in learning. *Neural Computation*, *10*, 251–276.

Amari, S., Chen, T. P., & Cichocki, A. (1997a). Stability analysis of learning algorithms for blind source separation. *Neural Networks*, *10*, 1345–1351.

Amari, S., Chen, T. P., & Cichocki, A. (2000). Nonholonomic orthogonal learning algorithms for blind source separation. *Neural Computation*, *12*, 1463–1484.

Amari, S., Douglas, S. C., Cichocki, A., & Yang, H. H. (1997b). Multichannel blind deconvolution and equalization using the natural gradient. *Proc. Signal Processing Advances in Wireless Communications* (pp. 101–104). Paris, France.

Attias, H., & Schreiner, C. E. (1998). Blind source separation and deconvolution: The dynamic component analysis algorithms. *Neural Computation*, *10*, 1373–1424.

Bartlett, M. S., Lades, H. M., & Sejnowski, T. J. (1998). Independent component representations for face recognition. *Proceedings of the SPIE Symposium on Electronic Imaging: Science and Technology; Conference on Human Vision and Electronic Imaging III* (pp. 528–539). San Jose, California.

Choi, S. (1998). Differential Hebbian-type learning algorithms for decorrelation and independent component analysis. *Electronics Letters*, *34*, 900–901.

Choi, S. (2002). Adaptive differential decorrelation: A natural gradient algorithm. *Proc. Int'l Conf. Artificial Neural Networks* (pp. 1168–1173). Madrid, Spain.

Choi, S. (2003). Differential learning and random walk model. *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing* (pp. 724–727). Hong Kong.

Choi, S., Amari, S., Cichocki, A., & Liu, R. (1999). Natural gradient learning with a nonholonomic constraint for blind deconvolution of multiple channels. *Proc. ICA'99* (pp. 371–376). Aussois, France.

Choi, S., Cichocki, A., & Amari, S. (2000). Flexible independent component analysis. *Journal of VLSI Signal Processing*, *26*, 25–38.

Choi, S., Cichocki, A., & Amari, S. (2002). Equivariant nonstationary source separation. *Neural Networks*, *15*, 121–130.

Choi, S., Cichocki, A., Park, H. M., & Lee, S. Y. (2005). Blind source separation and independent component analysis: A review. *Neural Information Processing - Letters and Review*, *6*, 1–57.

Choi, S., Cichocki, A., Zhang, L., & Amari, S. (2003). Approximate maximum likelihood source separation using the natural gradient. *IEICE Trans. Fundamentals*, *E86-A*, 206–214.

Choi, S., & Lee, O. (2000). Factorial code representation of faces for recognition. In S. W. Lee, H. H. Bülthoff and T. Poggio (Eds.), *Lecture Notes in Computer Science, 1811, Biologically Motivated Computer Vision*, 42–51. Springer.

Cichocki, A., & Amari, S. (2002). *Adaptive blind signal and image processing: Learning algorithms and applications*. John Wiley & Sons, Inc.

Földiák, P. (1989). Adaptive network for optimal linear feature extraction. *Proc. Int'l Joint Conf. Neural Networks* (pp. 401–405).

Földiák, P. (1990). Forming sparse representations by local anti-hebbian learning. *Biological Cybernetics*, *64*, 165–170.

Hebb, D. O. (1949). *The organization of behavior*. New York: Wiley.

Hyvärinen, A., Karhunen, J., & Oja, E. (2001). *Independent component analysis.* John Wiley & Sons, Inc.

Kosko, B. (1986). Differential Hebbian learning. *Proc. American Institute of Physics: Neural Networks for Computing* (pp. 277–282).

Lee, T. W., Girolami, M., Bell, A., & Sejnowski, T. (2000). A unifying information-theoretic framework for independent component analysis. *International Journal on Mathematical and Computer Modeling*, *39*, 1–21.

Linsker, R. (1988). Self-organization in a perceptual network. *Computer*, *21*, 105–117.

MacKay, D. J. C. (1996). *Maximum likelihood and covariant algorithms for independent component analysis* (Technical Report Draft 3.7). University of Cambridge, Cavendish Laboratory.

Matsuoka, K., Ohya, M., & Kawamoto, M. (1995). A neural net for blind separation of nonstationary signals. *Neural Networks*, *8*, 411–419.

Samaria, F., & Harter, A. (1994). Parameterisation of a stochastic model for human face identification. *Proc. 2nd IEEE Workshop on Applications of Computer Vision.* Sarasota, FL.

Sirovich, L., & Kirby, M. (1987). Low-dimensional procedure for the charaterization of human faces. *Journal of the Optical Society of America A*, *4*, 519–524.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *3*, 71–86.

List of Figures

1 Differential correlation between y_1 and y_2 in Example 1. 23

2 Evolution of performance index in Example 2: (a) conventional ICA; (b) differential ICA. 24

3 Hinton's diagram for the global matrix $\boldsymbol{G}$ in Example 2: (a) conventional ICA; (b) differential ICA. Each square's area represents the magnitude of the element of the matrix $\boldsymbol{G}$. White square is for positive sign and black square is for negative sign. 25

4 Original music signal and noise source used in Example 3. 26

5 Linear instantaneous mixtures of a music signal and a white Gaussian noise source in Example 3. 27

6 Music and noise signals recovered by ICA in Example 3. 28

7 Music and noise signals recovered by the differential ICA in Example 3. 29

8 Hinton diagrams for $\boldsymbol{G}$ in Example 3: (a) ICA; (b) differential ICA. 30

9 Example 4: Basis face images computed by: (a) ICA; (b) differential ICA. 31

List of Tables

1 Example 4: Performance comparison in face recognition using ORL face DB. For each method, 5-fold cross validation was applied and results in terms of percent correct are averaged hit rates. 32

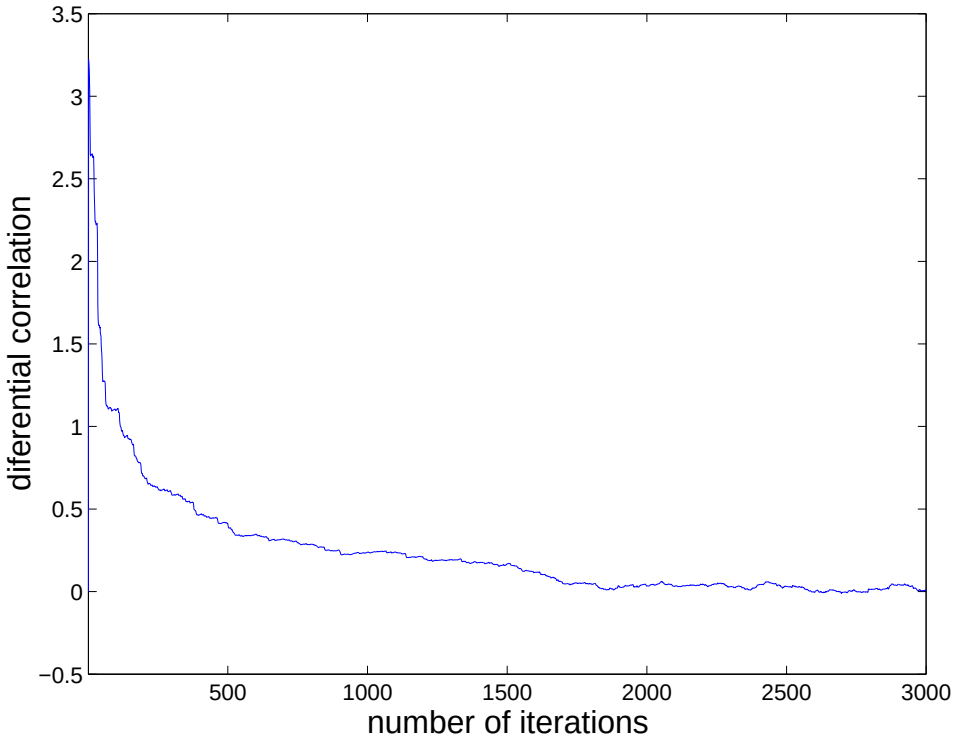
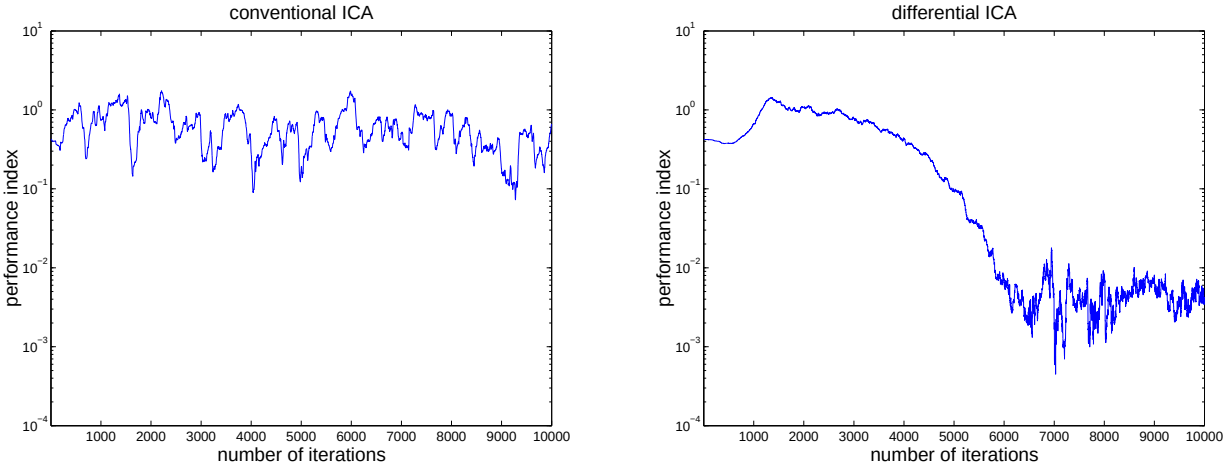
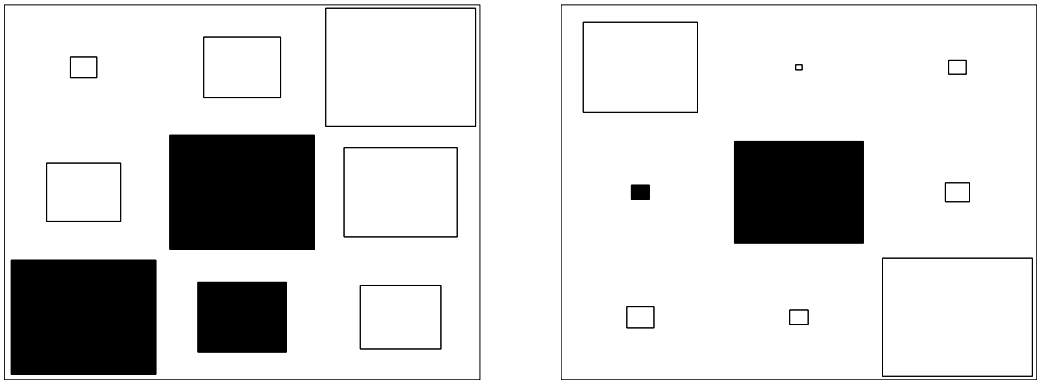


Figure 1: Differential correlation between y_1 and y_2 in Example 1.



(a) (b)

Figure 2: Evolution of performance index in Example 2: (a) conventional ICA; (b) differential ICA.



(a) (b)

Figure 3: Hinton's diagram for the global matrix $\boldsymbol{G}$ in Example 2: (a) conventional ICA; (b) differential ICA. Each square's area represents the magnitude of the element of the matrix $\boldsymbol{G}$. White square is for positive sign and black square is for negative sign.

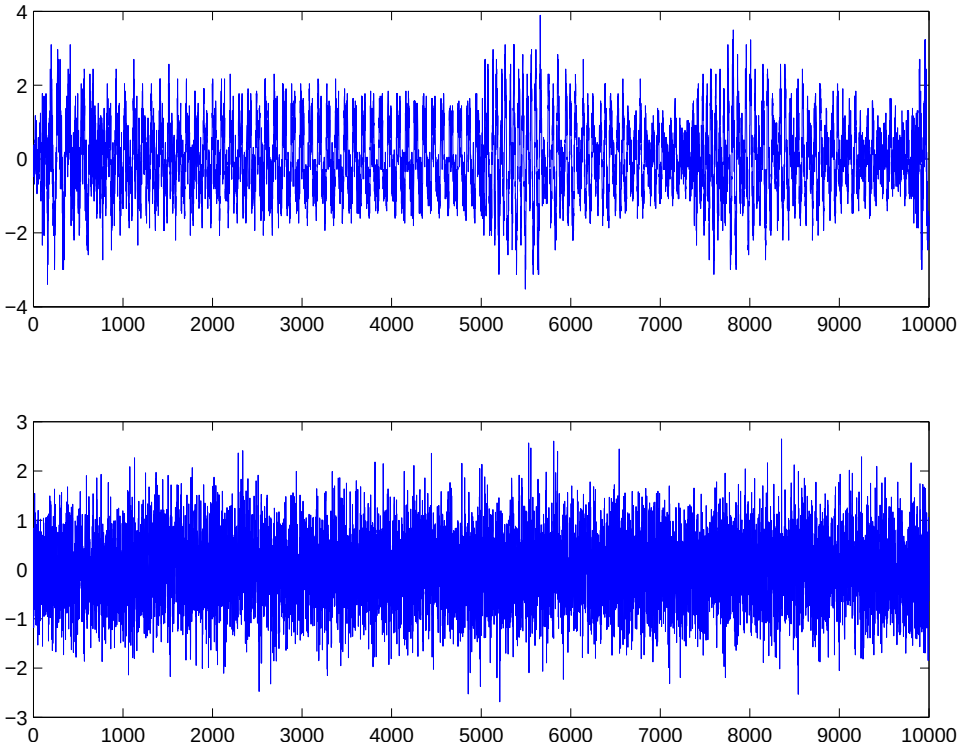


Figure 4: Original music signal and noise source used in Example 3.

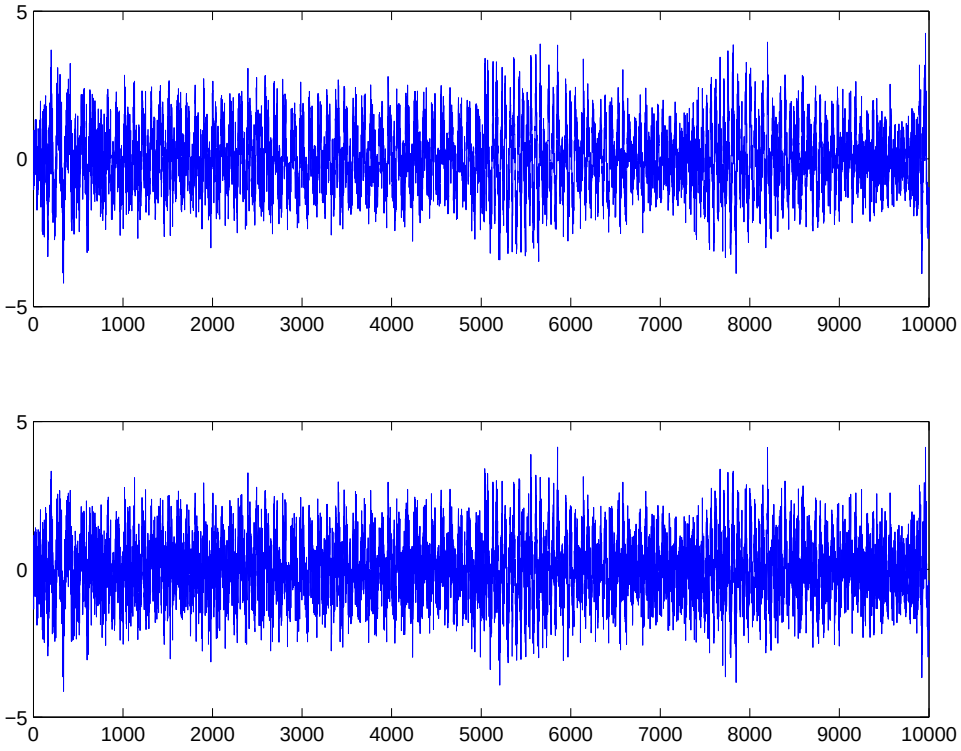


Figure 5: Linear instantaneous mixtures of a music signal and a white Gaussian noise source in Example 3.

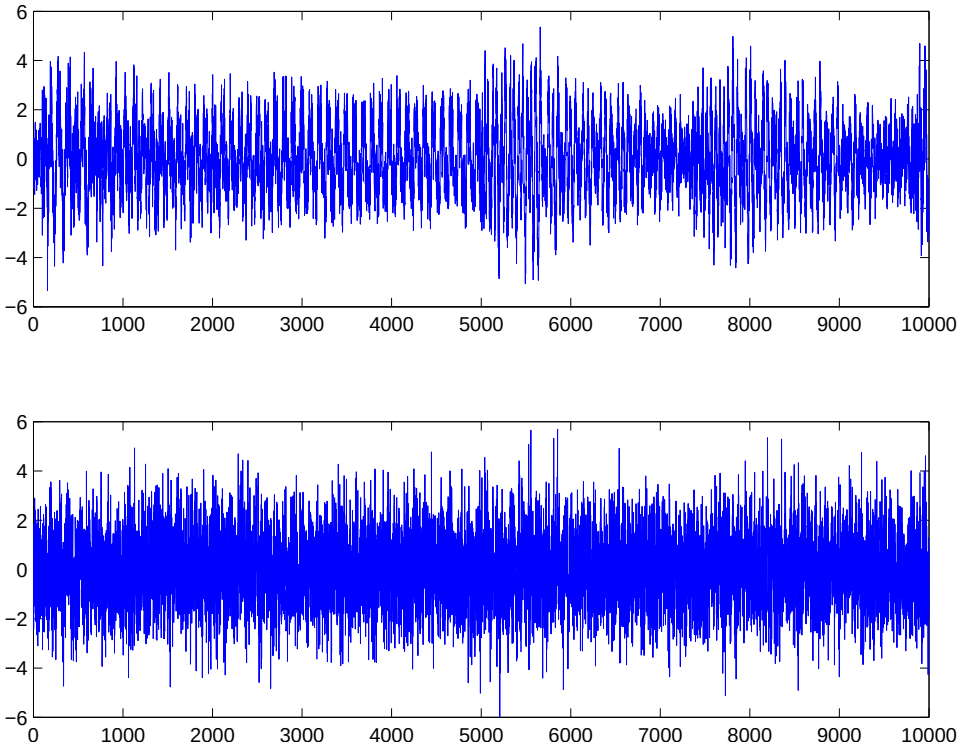


Figure 6: Music and noise signals recovered by ICA in Example 3.

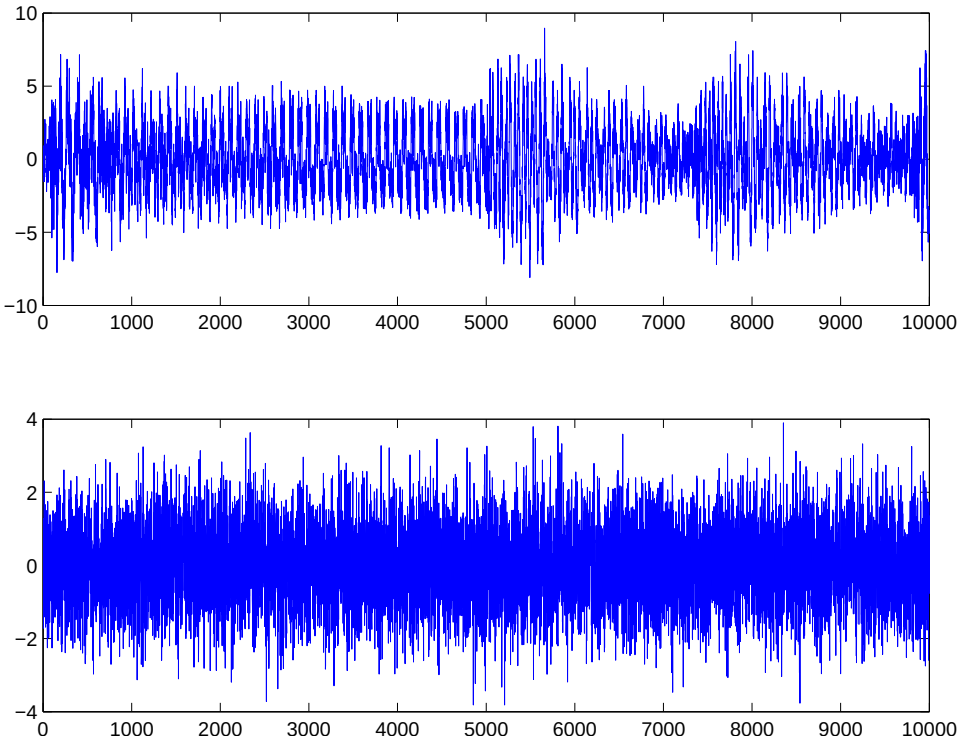
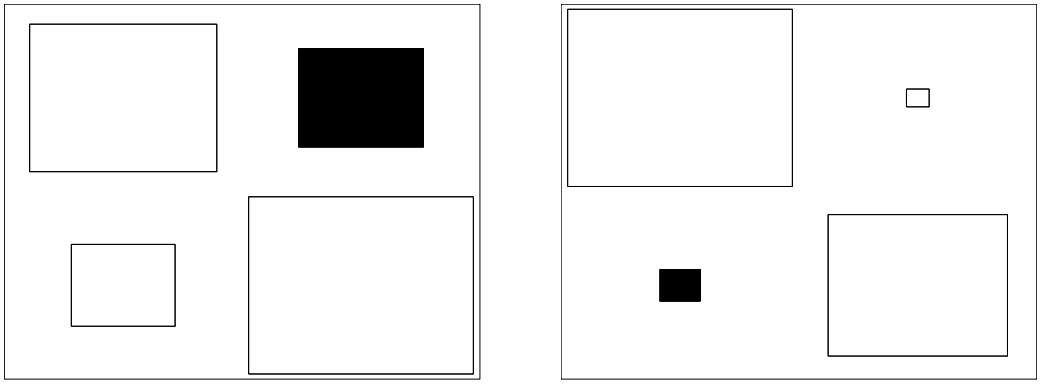
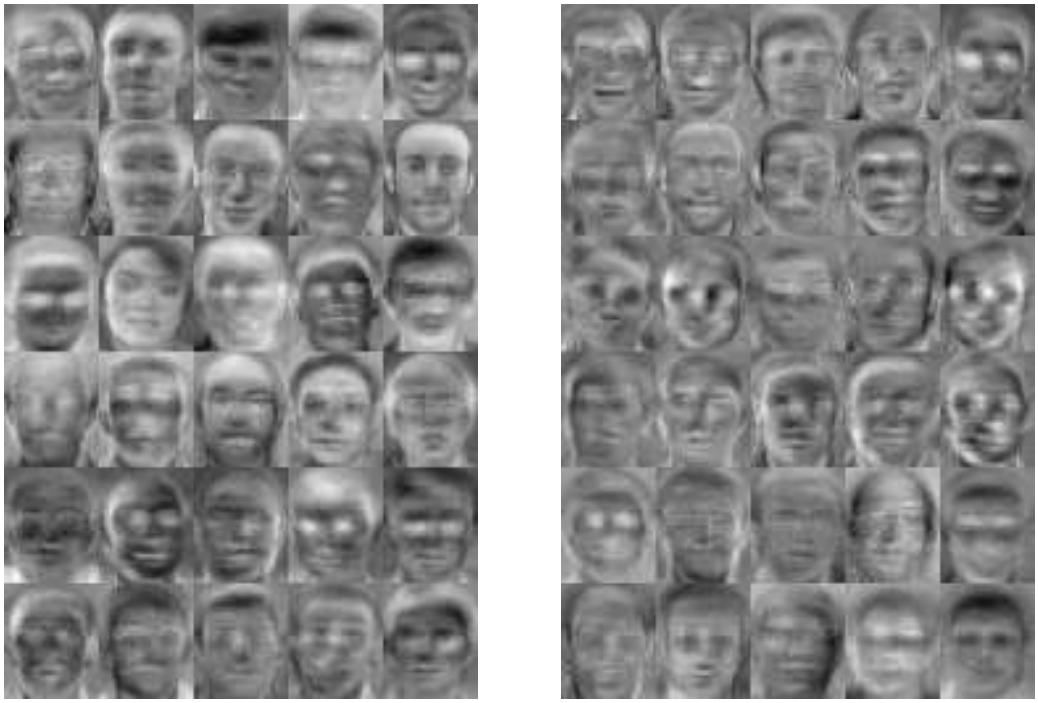


Figure 7: Music and noise signals recovered by the differential ICA in Example 3.



(a) (b)

Figure 8: Hinton diagrams for $\boldsymbol{G}$ in Example 3: (a) ICA; (b) differential ICA.



(a) (b)

Figure 9: Example 4: Basis face images computed by: (a) ICA; (b) differential ICA.

Table 1: Example 4: Performance comparison in face recognition using ORL face DB. For each method, 5-fold cross validation was applied and results in terms of percent correct are averaged hit rates.

Algorithm	Percent correct	# of misclassified face images
PCA	97.25%	$\frac{1+3+1+2+4}{5\times 80} = \frac{11}{400}$
ICA	95.60%	$\frac{2+4+2+6+4}{5\times 80} = \frac{18}{400}$
Differential ICA	**98.25**%	$\frac{1+3+0+1+2}{5\times 80} = \frac{7}{400}$