Differential ICA

Seungjin Choi

Department of Computer Science and Engineering
POSTECH
San 31 Hyoja-dong, Nam-gu
Pohang 790-784, Korea
seungjin@postech.ac.kr

Abstract. As an alternative to the conventional Hebb-type unsupervised learning, differential learning was studied in the domain of Hebb's rule [1] and decorrelation [2]. In this paper we present an ICA algorithm which employs differential learning, thus named as *differential ICA*. We derive a differential ICA algorithm in the framework of maximum likelihood estimation and random walk model. Algorithm derivation using the natural gradient and local stability analysis are provided. Usefulness of the algorithm is emphasized in the case of blind separation of temporally correlated sources and is demonstrated through a simple numerical example.

1 Introduction

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. In the framework of linear transform, ICA finds a representation of the form

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

where $\boldsymbol{x} \in \mathbb{R}^n$ is the observation data (which is given) and $\boldsymbol{A} = [\boldsymbol{a}_1 \cdots \boldsymbol{a}_n] \in \mathbb{R}^{n\times n}$ (which is known as a *mixing matrix* in source separation) consists of basis vectors $\{\boldsymbol{a}_i\}$ and $\boldsymbol{s} = [s_1 \cdots s_n]$ is an n-dimensional vector containing basis coefficients $\{s_i\}$ (which are called *independent components* and are also known as *sources*).

It is known that ICA performs source separation, the goal of which is to restore unknowns 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 *demixing matrix*).

Let $\boldsymbol{y}(t)$ be the output of demixing transform, i.e.,

$$\boldsymbol{y}(t) = \boldsymbol{W}\boldsymbol{x}(t). \tag{2}$$

Either maximum likelihood estimation or the minimization of mutual information leads to the well-known natural gradient ICA algorithm [3] whose updating rule has the form

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

where η is a learning rate and $\varphi(\boldsymbol{y}) = [\varphi_1(y_1) \cdots \varphi_n(y_n)]^T$ 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 [4, 5] (and references therein).

In a wide sense, most of ICA algorithms based on unsupervised learning belong to Hebb-type rule or its generalization with adopting nonlinear functions. Motivated from differential Hebb's rule [1] and differential decorrelation [2], we develop an ICA algorithm which employs differential learning where learning resorts to differentiated values (or difference of values in discrete-time counterpart).

Differential Hebb's rule was studied as an alternative to the Hebb's rule The motivation of the differential Hebb's rule is that concurrent change, rather than just concurrent activation, more accurately captures the *concomitant variation*. The differential learning was introduced in the framework of ICA [6] and decorrelation [2] recently . In this paper we derive a differential ICA algorithm in the framework of maximum likelihood estimation and random walk model. In fact, our differential ICA algorithm can be viewed as a simpler form of ICA algorithms which exploit the temporal structure of sources [7, 8].

2 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 that latent variables being statistically independent, is a semiparametric estimation problem. The maximum likelihood estimation of basis vectors $\{\boldsymbol{a}_i\}$ is involved with 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 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.

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

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

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

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

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

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

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

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

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

The factorial model given in (9) will be used as a optimization criterion to derive the proposed algorithm.

3 Differential ICA Algorithm

Denote a set of observation data by

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

where

$$\underline{\boldsymbol{x}}_i = \{x_i(0), \ldots, x_i(N-1)\}. \tag{11}$$

Then the normalized log-likelihood is given by

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

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.

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

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

Note that objective function (14) is slightly different from the one 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 [3, 9, 10]. 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 $\varphi(\boldsymbol{y}') = [\varphi_1(y_1')\cdots\varphi_n(y_n')]^T$.

With this definition, we have

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

One can easily see that

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

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

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

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

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

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

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

Remarks

- The algorithm (22) was originally derived in an *ad hoc* manner in [6]. 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 temporal ICA algorithm [7] 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 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 [10], we can adopt a flexible nonlinear function based on the generalized Gaussian distribution.

4 Local Stability Analysis

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 similarly, following the result in [3]. As in [3], 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

$$\begin{aligned} E\{d^2\mathcal{J}_3\} &= E\left\{\boldsymbol{y}' d\boldsymbol{V}^T \boldsymbol{\Phi} d\boldsymbol{y}' + \varphi^T(\boldsymbol{y}') d\boldsymbol{V} d\boldsymbol{y}'\right\} \\ &= E\left\{\boldsymbol{y}' d\boldsymbol{V}^T \boldsymbol{\Phi} d\boldsymbol{V}\boldsymbol{y}' + \varphi^T(\boldsymbol{y}') d\boldsymbol{V} d\boldsymbol{V}\boldsymbol{y}'\right\} \\ &= \sum_{j\neq i}\left[\sigma_i^2 \kappa_j \left(dv_{ji}\right)^2 + dv_{ij} dv_{ji}\right] + \sum_i (\zeta_i + 1)\left(dv_{ii}\right)^2, \end{aligned} \qquad (23)$$

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

$$\boldsymbol{\Phi} = \begin{bmatrix} \dot{\varphi}_1(y_1') & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \dot{\varphi}_n(y_n') \end{bmatrix} \qquad (24)$$

$$\dot{\varphi}_i(y_i') = \frac{d\varphi_i(y_i')}{dy_i'} \qquad (25)$$

$$\sigma_i^2 = E\{y_i'^2\} \qquad (26)$$

$$\kappa_i = E\{\dot{\varphi}_i(y_i')\} \qquad (27)$$

$$\zeta_i = E\{y_i'^2 \dot{\varphi}_i(y_i')\}. \qquad (28)$$

It follows from (23) that $E\{d^2\mathcal{J}_3\}$ is positive if and only if

$$\kappa_i > 0 \qquad (29)$$

$$\zeta_i + 1 > 0 \qquad (30)$$

$$\sigma_i^2\sigma_j^2\kappa_i\kappa_j > 1. \qquad (31)$$

5 Numerical Example

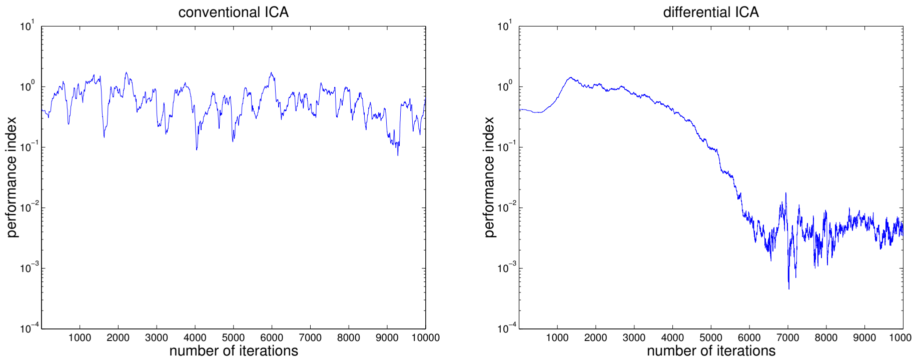
We present a simple numerical example to show the usefulness of our differential ICA algorithm which is described in (22). Three independent innovation sequences were drawn from Laplacian distribution. Each innovation sequence was convolved with a moving average filter (with exponentially decreasing 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

$$\mathrm{PI} = \frac{1}{2(n-1)} \sum_{i=1}^{n} \left\{ \left(\sum_{k=1}^{n} \frac{|g_{ik}|^2}{\max_j |g_{ij}|^2} - 1 \right) + \left(\sum_{k=1}^{n} \frac{|g_{ki}|^2}{\max_j |g_{ji}|^2} - 1 \right) \right\}, \qquad (32)$$

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 (32) 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. 1.



(a)

(b)

Fig. 1. Evolution of performance index: (a) conventional ICA; (b) differential ICA.

6 Discussion

In this paper we have presented a natural gradient learning algorithm for differential decorrelation, the goal of which is to minimize the correlation between differentiated random variables. We showed that the differential decorrelation algorithm could be derived from learning a linear generative model by the maximum likelihood estimation under a random walk model. We also discussed a differential version of the natural gradient ICA algorithm and showed that it could also be derived under the random walk model. The differential correlation algorithm (22) or the differential ICA algorithm (22) could be generalized by adopting higher-order differentiation. This generalization is currently under investigation.

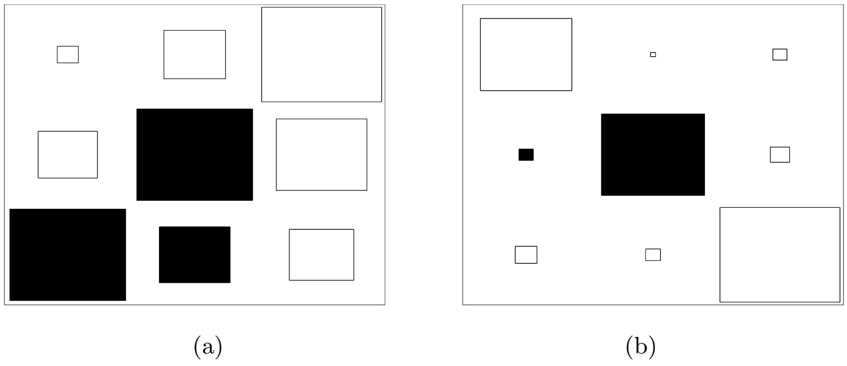


Fig. 2. Hinton's diagram for the global matrix $\boldsymbol{G}$: (a) conventional ICA; (b) differential ICA. Each square's area represents the magnitude of the element of the matrix $\boldsymbol{G}$. White square is for positive sign and black square is for negative sign.

7 Acknowledgments

This work was supported by Korea Ministry of Science and Technology under Brain Science and Engineering Research Program and an International Cooperative Research Project, by KOSEF 2000-2-20500-009-5, and by Ministry of Education of Korea for its financial support toward the Electrical and Computer Engineering Division at POSTECH through its BK21 program.

References

1. Kosko, B.: Differential Hebbian learning. In: Proc. American Institute of Physics: Neural Networks for Computing. (1986) 277–282
2. Choi, S.: Adaptive differential decorrelation: A natural gradient algorithm. In: Proc. ICANN, Madrid, Spain (2002) 1168–1173
3. Amari, S., Chen, T.P., Cichocki, A.: Stability analysis of learning algorithms for blind source separation. Neural Networks **10** (1997) 1345–1351
4. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. John Wiley & Sons, Inc. (2001)
5. Cichocki, A., Amari, S.: Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications. John Wiley & Sons, Inc. (2002)
6. Choi, S.: Differential Hebbian-type learning algorithms for decorrelation and independent component analysis. Electronics Letters **34** (1998) 900–901
7. Attias, H., Schreiner, C.E.: Blind source separation and deconvolution: The dynamic component analysis algorithms. Neural Computation **10** (1998) 1373–1424
8. Amari, S.: Estimating functions of independent component analysis for temporally correlated signals. Neural Computation **12** (2000) 2083–2107
9. Amari, S.: Natural gradient works efficiently in learning. Neural Computation **10** (1998) 251–276
10. Choi, S., Cichocki, A., Amari, S.: Flexible independent component analysis. Journal of VLSI Signal Processing **26** (2000) 25–38