

# STABILITY ANALYSIS OF ADAPTIVE BLIND SOURCE SEPARATION

Shun-ichi AMARI \*, Tian-Ping CHEN, Andrzej CICHOCKI

RIKEN Frontier Research Program

Brain Information Processing Group

## Abstract

Recently a number of adaptive learning algorithms have been proposed for blind source separation. Although the underlying principles and approaches are different, most of them have very similar forms. Two important issues have remained to be elucidated further: the statistical efficiency and the stability of learning algorithms. The present letter analyzes a general form of statistically efficient algorithm and give a necessary and sufficient condition for the separating solution to be a stable equilibrium of a general learning algorithm. Moreover, when the separating solution is unstable, a simple method is given for stabilizing the separating solution by modifying the algorithm.

**Key words:** — Blind source separation, Stability of learning, Efficiency of learning, Natural gradient, Independent component analysis (ICA).

---

\*Requests for reprints should be sent to Shun-ichi Amari, RIKEN Frontier Research Program, Hirosawa 2-1, Wako-shi, Saitama 351-01, Japan ; amari@zoo.riken.go.jp

# 1 Introduction

Blind source separation is the problem of recovering a number of original stochastic independent signals when only their linear mixtures are available. Here, “blind” implies that neither the mixing coefficients nor the probability distributions of the original source signals are known. This problem is important in the field of biological signal processing (e.g., MEG, ECG and EEG data) as well as communication engineering. Being inspired from a neurobiological perspective, Jutten and Herault ( Jutten, Herault, 1986 and 1991) proposed a heuristic learning algorithm to solve this problem. However, their algorithm has rather poor performance for more than mixture of two sources. Since then, a lot of new ideas have been proposed by many researchers. These include the independent component analysis (ICA )(Common, 1994), nonlinear principal component analysis (Oja, Kahrunen, 1995), entropy maximization (Bell and Sejnowski, 1995), robust adaptive algorithm (Cichocki et al. 1994), equivariant adaptive algorithm (Cardoso and Laheld, 1996), and the natural gradient approach (Amari et al. 1995).

There are two important theoretical problems to be elucidated; namely the efficiency (performance) and the stability of proposed learning algorithms. The efficiency is investigated by Amari and Cardoso (1997) by using semi-parametric statistical models and estimating functions. The stability of a learning algorithm was analyzed by Fetty (Fetty 1991), Sorouchyari (Sorouchyari, 1991), Chen (Chen 1994), Macchi and Moreau (Macchi, Moreau 1996) and Deville (Deville,1996) for the special case where the number of sources is two. Recently Cardoso and Laheld (Cardoso and Laheld 1996) gave stability analysis in the case of  $m$  sources where skew symmetric components are major parts of the learning algorithm.

The present letter gives a general stability analysis (in the sense given by Sorouchyari and others) for learning algorithms of blind source separation. The natural (nonholonomic) gradient basis is used to elucidate the stability of stochastic descent learning algorithms. The stability of a given algorithm depends on the unknown stochastic natures of the source signals. The conditions of stability are explicitly given in terms of statistical parameters. Moreover, we propose a general method to transform an unstable learning algorithm into a stable and efficient one.

In this paper we use the following standard notations. The vectors and matrices are denoted by bold small and capital letters, respectively, e.g. a vector  $\mathbf{s}$  and matrix  $\mathbf{A}$ . The  $i$ -th element of vector  $\mathbf{s}$  is denoted by  $s_i$  and the  $ij$ -th element of a matrix  $\mathbf{A}$  by  $a_{ij}$ . The superscript  $T$  denotes the transpose of a matrix or vector. Time is denoted by  $t$  which can take continuous or discrete values  $(0, 1, 2, \dots)$ .  $E[s]$  denotes the expectation of a random variable  $s$ .

## 2 Problem Formulation

Let us consider  $m$  independent and stationary sources signals, which are denoted by a column vector  $\mathbf{s} = [s_1, \dots, s_n]^T$ . The sources generate signals  $\mathbf{s}(1), \mathbf{s}(2), \dots, \mathbf{s}(t), \dots$ , at discrete time. We say that signals are independent if their joint probability distribution function can be decomposed as

$$p(\mathbf{s}) = \prod_{i=1}^m p_i(s_i), \quad (2.1)$$

where  $p_i(s_i)$  is the pdf (probability density function) of  $i$ th source signal. We also assume that the source signals are zero mean, although these assumptions are easily relaxed. Hence, we have

$$E[s_i(t)] = 0, \quad (2.2)$$

where  $E$  denotes the expectation

We assume that  $m$  mixed signals  $\mathbf{x} = [x_1, \dots, x_m]^T$  are observed instead of  $\mathbf{s}$ . Here,  $\mathbf{x}(t)$  consists of instantaneous mixtures of  $\mathbf{s}(t)$

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t), \quad (2.3)$$

where  $\mathbf{A}$  is an unknown  $m$  by  $m$  nonsingular mixing matrix not depending on  $t$ . The problem is to recover original source signals  $\mathbf{s}(t)$  from observations  $\mathbf{x}(t)$  ( $t = 1, 2, \dots$ ). If we obtain a good estimator  $\mathbf{W}(t)$  of  $\mathbf{A}^{-1}$  based on  $T$  observations  $\mathbf{x}(1), \dots, \mathbf{x}(T)$ , the signals are recovered by

$$\mathbf{y}(t) = \mathbf{W}(t)\mathbf{x}(t). \quad (2.4)$$

This can be realized by a feed-forward linear neural network with connection weight matrix

$\mathbf{W}$  or by a feedback (fully recurrent) neural network described as

$$\mathbf{y}(t) = \mathbf{x}(t) - \mathbf{V}(t)\mathbf{y}(t), \quad (2.5)$$

with feedback connection matrix  $\mathbf{V}(t)$ , The both networks are equivalent when

$$\mathbf{W}(t) = [\mathbf{I} + \mathbf{V}(t)]^{-1}, \quad (2.6)$$

where  $\mathbf{I}$  is the identity matrix.

The estimator  $\mathbf{W}(t)$  can be obtained by a neural learning algorithm of the following general form

$$\mathbf{W}(t+1) = \mathbf{W}(t) - \eta(t)\mathbf{F}[\mathbf{y}(t), \mathbf{W}(t)] \quad (2.7)$$

for feed-forward network and

$$\mathbf{V}(t+1) = \mathbf{V}(t) - \eta(t)[\mathbf{V}(t) + \mathbf{I}]\mathbf{F}[\mathbf{y}(t), \mathbf{V}(t)] \quad (2.8)$$

for fully recurrent network (Amari et al. 1995), where  $\eta(t)$  is a positive learning rate and  $\mathbf{F}$  is a matrix function which determines the learning algorithm. In general,  $\mathbf{F}$  can be represented as a function of inputs  $\mathbf{x}(t)$ , outputs  $\mathbf{y}(t)$  and weights  $\mathbf{W}(t)$ , but it can be rewritten as a function of only outputs  $\mathbf{y}(t)$  and synaptic weights  $\mathbf{W}(t)$ .

It should be noted, however, that the matrix  $\mathbf{A}$  (or its inverse) is not identifiable from the observed signals. Even if we can extract  $m$  independent signals, we do not know their ordering. This implies that there exists a freedom of permutations of the original signals. The magnitudes of the original signals  $s_i$  are also not recoverable, because a scalar multiple of  $s_i$ ,  $cs_i$  of  $s_i$  by a constant  $c$  cannot be distinguished from multiplication of the  $i$ th column of  $\mathbf{A}$  by the same constant  $c$ . Therefore, we can recover only a permuted and rescaled version of the original signals. This means that we can obtain  $\mathbf{W} = \mathbf{PDA}^{-1}$  at best, where  $\mathbf{P}$  is permutation matrix and  $\mathbf{D}$  is nonsingular scaling matrix. For simplicity, in our further consideration we assume that  $\mathbf{PD} = \mathbf{I}$ . In order to resolve indefiniteness of scales, without loss of generality we impose the following normalization (scaling) conditions on all  $s_i$

$$E[f_i(s_i)] = 0, \quad (2.9)$$

where  $f_i(s_i)$  is a suitable nonlinear function. For example, when  $f_i(s_i) = s_i^2 - 1$  the variance  $\sigma_i^2$  of  $s_i$  is assumed to be equal to 1. However, it is possible to have any other normalizing conditions such as  $f_i(s) = s_i^4 - 1$ .

### 3 Loss functions and equivariant learning algorithms

Most learning algorithms are derived from heuristic considerations (Jutten and Herault, 1986), or are based on minimization or maximization of a loss or performance function. Entropy maximization (Bell and Sejnowski, 1995), ICA (Common, 1994; Amari et al., 1995) and maximization of likelihood lead to the same type of loss function,

$$l(\mathbf{y}, \mathbf{W}) = -\log |\det(\mathbf{W})| - \sum_{i=1}^m \log p_i(y_i), \quad (3.1)$$

where  $p_i(y_i)$  are probability density functions (pdf) of output signals and  $|\det(\mathbf{W})|$  means the absolute value of determinant of matrix  $\mathbf{W}$ . We can then apply the stochastic gradient descent learning method to derive the learning rule. We will discuss a more general or universal type of algorithms in the next section.

In order to calculate the gradient of  $l$ , we derive the total differential  $dl$  of  $l$  when  $\mathbf{W}$  is changed from  $\mathbf{W}$  to  $\mathbf{W} + d\mathbf{W}$ . In the component form,

$$dl = l(\mathbf{y}, \mathbf{W} + d\mathbf{W}) - l(\mathbf{y}, \mathbf{W}) = \sum_{i,j} \frac{\partial l}{\partial w_{ij}} dw_{ij}, \quad (3.2)$$

where the coefficients  $\partial l / \partial w_{ij}$  of  $dw_{ij}$  represent the gradient of  $l$ .

Simple algebraic and differential calculus yields

$$dl = -\text{tr}(d\mathbf{W}\mathbf{W}^{-1}) + \varphi(\mathbf{y})^T d\mathbf{y}, \quad (3.3)$$

where  $\text{tr}$  is the trace of a matrix and  $\varphi(\mathbf{y})$  is a column vector whose components are

$$\varphi_i(y_i) = -\frac{\dot{p}_i(y_i)}{p_i(y_i)}, \quad (3.4)$$

where “ $\dot{\cdot}$ ” denoting the differentiation. From  $\mathbf{y} = \mathbf{W}\mathbf{x}$ , we have

$$d\mathbf{y} = d\mathbf{W}\mathbf{x} = d\mathbf{W}\mathbf{W}^{-1}\mathbf{y}. \quad (3.5)$$

Hence, we put

$$d\mathbf{X} = d\mathbf{W}\mathbf{W}^{-1}, \quad (3.6)$$

whose components  $dx_{ij}$  are linear combinations of  $dw_{ij}$ . The differentials  $dx_{ij}$  form a basis of the tangent space of nonsingular matrices  $\mathbf{W}$  since they are linear combinations of the basis  $dw_{ij}$ . It should be noted that  $d\mathbf{X} = d\mathbf{W}\mathbf{W}^{-1}$  is a non-integrable differential form so that we do not have a matrix function  $\mathbf{X}(\mathbf{W})$  which gives (3.6). Nevertheless, the nonholonomic basis  $d\mathbf{X}$  has a definite geometrical meaning and is very useful. It is effective to analyze the differential in terms of  $d\mathbf{X}$ , since the natural Riemannian gradient (Amari et al., 1995; Amari, 1997) is automatically implemented by it and the equivariant properties investigated by Cardoso and Laheld (1996) automatically hold in this basis. It is easy to rewrite the results in terms of  $d\mathbf{W}$  by using (3.6).

The gradient  $dl$  in (3.3) is expressed in the differential form

$$dl = -\text{tr}(d\mathbf{X}) + \boldsymbol{\varphi}(\mathbf{y})^T d\mathbf{X} \mathbf{y}. \quad (3.7)$$

This leads to the stochastic gradient learning algorithm,

$$\Delta\mathbf{X}(t) = \mathbf{X}(t+1) - \mathbf{X}(t) = -\eta(t) \frac{dl}{d\mathbf{X}} = \eta(t) \left[ \mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}(t))\mathbf{y}^T(t) \right]$$

in terms of  $\Delta\mathbf{X}(t) = \Delta\mathbf{W}(t)\mathbf{W}^{-1}(t)$ , or

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \eta(t) \left[ \mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}(t))\mathbf{y}^T(t) \right] \mathbf{W}(t) \quad (3.8)$$

in terms of  $\mathbf{W}(t)$ .

Using Eqs. (2.5) and (2.6) we could derive similar algorithm for fully recurrent neural network as

$$\mathbf{V}(t+1) = \mathbf{V}(t) + \eta(t) [\mathbf{V}(t) + \mathbf{I}] \left[ \mathbf{I} - \boldsymbol{\varphi}(\mathbf{y}(t))\mathbf{y}^T(t) \right]. \quad (3.9)$$

The above learning algorithms are equivariant because their asymptotic convergence properties are independent of scaling factors and condition number of the mixing matrix  $\mathbf{A}$ .

## 4 General learning algorithms and estimating functions

Here, we give a statistical consideration of the problem, and show the class of efficient learning algorithms. We then justify the use of a loss function, because, even though efficient algorithms are not necessarily of the gradient form, they are derived therefrom.

The probability density function of  $\mathbf{x}$  can be expressed as (Amari and Cardoso, 1997)

$$p_X[\mathbf{x}, \mathbf{W}, p(\mathbf{s})] = |\det(\mathbf{W})|p(\mathbf{W}\mathbf{x}), \quad (4.1)$$

which depends on two unknowns  $\mathbf{W} = \mathbf{A}^{-1}$  and  $p(\mathbf{s})$ .

Here, the statistical model (4.1) includes not only an unknown matrix  $\mathbf{W}$  to be estimated but also an unknown function  $p(\mathbf{s})$  which we do not need to estimate. Such a model is said to be semi-parametric, and is a difficult statistical problem (see, for example, Bickel et al, 1993).

Let  $\mathbf{F}(\mathbf{y}, \mathbf{W})$  be a matrix satisfying

$$E[\mathbf{F}(\mathbf{y}, \mathbf{W})] = \mathbf{o}, \quad (4.2)$$

where the expectation is taken with respect to  $\mathbf{y}$  by using any probability density function  $p(\mathbf{s})$  of form (4.1). Such a matrix which does not depend on  $p(\mathbf{s})$  is called an estimating function, provided it satisfies a certain regularity conditions which we do not state here (Godambe, 1997 ; Amari and Kawanabe, 1996a). Once an estimating function is found, replacing the expectation by the sample average, an estimator  $\widehat{\mathbf{W}}$  is obtained by solving the estimating equation

$$\sum_{t=1}^T \mathbf{F}[\mathbf{y}(t), \mathbf{W}] = \mathbf{o}. \quad (4.3)$$

Here, it is not necessary to know  $p(\mathbf{s})$ . A learning algorithm is also derived from an estimating function as

$$\mathbf{W}(t+1) = \mathbf{W}(t) - \eta(t)\mathbf{F}[\mathbf{y}(t), \mathbf{W}(t)]. \quad (4.4)$$

Amari and Kawanabe (1996 a, b) proposed the information geometric theory of estimating functions by extending the differential geometry of statistics (Amari, 1985; Murray and Rice, 1993), and gave the set of all the estimating functions. The set is decomposed into the orthogonal sum of effective parts and ancillary parts. The ancillary parts reduce the efficiency

of estimators. The theory has been applied to blind source separation by Amari and Cardoso (1997) to give the Fisher efficient estimator. It is also proved that the effective part of the off-diagonal elements of estimating functions  $F_{ij}$  for  $(i \neq j)$  is spanned by the functions of the form  $\varphi(y_i)y_j$  and  $\varphi(y_j)y_i$  and the diagonal part by  $f(y_i) = \psi(y_i)y_i - 1$ , where  $\varphi$  and  $\psi$  are arbitrary functions.

From the theory of estimating functions, we thus have the general form of effective learning algorithms

$$\mathbf{F}(\mathbf{y}, \mathbf{W}) = K(\mathbf{W}) \circ [\mathbf{I} - \varphi(\mathbf{y})\mathbf{y}^T]\mathbf{W}, \quad (4.5)$$

where  $\varphi(\mathbf{y}) = [\varphi_1(y_1), \dots, \varphi_m(y_m)]^T$  is an arbitrary vector function and  $K(\mathbf{W})$  is an arbitrary linear operator which maps matrices to matrices. This is because a linear transform  $K(\mathbf{W}) \circ \mathbf{F}(\mathbf{y}, \mathbf{W})$  of  $\mathbf{F}$  gives again a statistically equivalent estimating function. Here, the normalization conditions are

$$E[f(y_i)] = E[\varphi_i(y_i)y_i - 1] = 0. \quad (4.6)$$

Some researchers have suggested to use a more general form like  $\varphi(\mathbf{y})^T \psi(\mathbf{y})$  to increase flexibility and/or improve the efficiency. However, estimating function theory proves that such a general function causes a further loss of efficiency, and the optimal one is found in the class of linear combinations of the form  $\varphi_i(y_i)y_j$  (see Amari and Cardoso 1997). The optimal one is to choose

$$\varphi_i(y) = -\frac{d}{dy_i} \log p_i(y_i) = -\frac{\dot{p}_i(y_i)}{p_i(y_i)},$$

if we can estimate the true source probability distribution  $p_i(s_i)$  adaptively. However, even when  $\varphi_i(y_i)$  is different from the above, the estimator is still consistent.

## 5 Stability of learning algorithms

For the moment we set  $K(\mathbf{W})$  be equal to the identity operator, giving the general form of effective learning algorithms

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \eta(t) \left[ \mathbf{I} - \varphi(\mathbf{y}(t))\mathbf{y}(t)^T \right] \mathbf{W}(t).$$

Alternatively, we use the continuous time version of the algorithm as

$$\dot{\mathbf{W}}(t) = \mu(t) \left[ \mathbf{I} - \varphi(\mathbf{y}(t))\mathbf{y}^T(t) \right] \mathbf{W}(t), \quad (5.1)$$

where  $\dot{\mathbf{W}}$  denotes time derivative of the matrix  $\mathbf{W}(t)$  and  $\mu(t) = \eta(t)/\tau$  and  $\tau$  is sampling period.

This kind of a stochastic gradient descent learning equation has been proposed by Amari (1967) for learning of neural networks including hidden units (later rediscovered and termed as the generalized delta rule). The dynamical aspects of on-line learning equations were studied also by Amari (1967).

The equilibrium points of the equation satisfy

$$E[\mathbf{I} - \varphi(\mathbf{y}(t))\mathbf{y}^T(t)] = \mathbf{0}, \quad (5.2)$$

which obviously has a solution  $\mathbf{W} = \mathbf{A}^{-1}$  (more precisely its permuted and rescaled version). However, this does not guarantee that  $\mathbf{W}(t)$  converges to  $\mathbf{A}^{-1}$  even locally.

We consider the expected version of the learning equation

$$\dot{\mathbf{W}}(t) = \mu(t)E[\mathbf{I} - \varphi(\mathbf{y}(t))\mathbf{y}^T(t)]\mathbf{W}(t) \quad (5.3)$$

By linearizing it at the equilibrium point, we have the variational equation

$$\delta\dot{\mathbf{W}}(t) = \mu(t) \frac{\partial \left( E \left[ \mathbf{I} - \varphi(\mathbf{y}(t))\mathbf{y}^T(t) \right] \mathbf{W} \right)}{\partial \mathbf{W}} \delta\mathbf{W}. \quad (5.4)$$

This shows that, only when all the eigenvalues of the operator  $\partial \left( E \left[ \mathbf{I} - \varphi(\mathbf{y})\mathbf{y}^T \right] \mathbf{W} \right) / \partial \mathbf{W}$  have negative real parts, the equilibrium is asymptotically stable. Therefore, we need to evaluate all the eigenvalues of the operator.

This can be done in terms of  $d\mathbf{X}$ , as follows.

Since  $\mathbf{I} - \varphi(\mathbf{y})\mathbf{y}^T$  is derived from the gradient  $dl$ , we need to calculate its Hessian  $d^2l$

$$d^2l = \sum \frac{\partial l(\mathbf{y}, \mathbf{W})}{\partial w_{ij} \partial w_{kl}} dw_{ij} dw_{kl}$$

in terms of  $d\mathbf{X}$ . The equilibrium is stable if and only if the expectation of the above quadratic form is positive definite.

Now, we can state the stability conditions explicitly. To this end, we put

$$\sigma_i^2 = E[y_i^2], \quad (5.5)$$

$$k_i = E[\dot{\varphi}_i(y_i)], \quad (5.6)$$

$$m_i = E[y_i^2 \dot{\varphi}_i(y_i)], \quad (5.7)$$

where  $\dot{\varphi} = d\varphi/dy$ .

**Theorem 1.** The separating solution is a stable equilibrium of learning equation (5.1), if and only if

$$m_i + 1 > 0, \quad (5.8)$$

$$k_i > 0, \quad (5.9)$$

$$\sigma_i^2 \sigma_j^2 k_i k_j > 1, \quad (5.10)$$

for all  $i, j$  ( $i \neq j$ ).

**Proof.** We calculate the second total differential, which is the quadratic form of the Hessian of  $l$ , as

$$\begin{aligned} d^2 l &= \mathbf{y}^T d\mathbf{X}^T \dot{\varphi}(\mathbf{y}) d\mathbf{y} + \varphi(\mathbf{y})^T d\mathbf{X} d\mathbf{y} \\ &= \mathbf{y}^T d\mathbf{X}^T \dot{\varphi}(\mathbf{y}) d\mathbf{X} \mathbf{y} + \varphi(\mathbf{y})^T d\mathbf{X} d\mathbf{X} \mathbf{y}, \end{aligned} \quad (5.11)$$

where  $\dot{\varphi}(\mathbf{y})$  is the diagonal matrix whose diagonal elements are  $\dot{\varphi}_1(y_1), \dots, \dot{\varphi}_m(y_m)$ .

The expectation of the first term is

$$\begin{aligned} E[\mathbf{y}^T d\mathbf{X}^T \dot{\varphi}(\mathbf{y}) d\mathbf{X} \mathbf{y}] &= \sum E[y_i dx_{ji} \dot{\varphi}_j(y_j) dx_{jk} y_k] \\ &= \sum_{j \neq i} E[(y_i)^2] E[\dot{\varphi}_j(y_j)] (dx_{ji})^2 + \sum_i E[(y_i)^2 \dot{\varphi}_i(y_i)] (dx_{ii})^2 \end{aligned} \quad (5.12)$$

$$= \sum_{j \neq i} \sigma_i^2 k_j (dx_{ji})^2 + \sum_i m_i (dx_{ii})^2. \quad (5.13)$$

Here, expectation is taken at  $\mathbf{W} = \mathbf{A}^{-1}$  so that  $y_i$ 's are independent.

Similarly,

$$E[\varphi(\mathbf{y})^T d\mathbf{X} d\mathbf{X} \mathbf{y}] = \sum E[\varphi(y_i) dx_{ij} dx_{jk} y_k]$$

$$= \sum E[y_i \varphi(y_i)] dx_{ij} dx_{ji} \quad (5.14)$$

$$= \sum_{i,j} dx_{ij} dx_{ji}, \quad (5.15)$$

because  $E[y_i \varphi(y_i)] = 1$  (the normalization condition).

Hence,

$$E[d^2 l] = \sum_{j \neq i} \{ \sigma_i^2 k_j (dx_{ji})^2 + dx_{ij} dx_{ji} \} \quad (5.16)$$

$$+ \sum_i (m_i + 1) (dx_{ii})^2. \quad (5.17)$$

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

$$q_{ij} = \sigma_i^2 k_j (dx_{ji})^2 + \sigma_j^2 k_i (dx_{ij})^2 + 2 dx_{ij} dx_{ji}. \quad (5.18)$$

This  $q_{ij} (i \neq j)$  is the quadratic form in  $(dx_{ij}, dx_{ji})$ , and

$$E[d^2 l] = \sum_{i \neq j} q_{ij} + \sum (m_i + 1) dx_{ii}. \quad (5.19)$$

The  $q_{ij}$  is positive if and only if (5.8) - (5.10) hold.

## 6 Illustrative examples

We illustrate practicability and universality of the derived stability conditions (5.8)-(5.10) by two simple examples.

Example 1

Let us consider the following odd activation function

$$\varphi_i(y_i) = |y_i|^p \operatorname{sgn}(y_i) \quad (6.1)$$

for  $p = 1, 2, 3, \dots$ . It is easy to check that the conditions (5.8)-(5.9) are always satisfied since

$$m_i = E[y_i^2 \dot{\varphi}_i(y_i)] = p E[|y_i|^{p+1}] > 0, \quad (6.2)$$

$$k_i = E[\dot{\varphi}_i(y_i)] = p E[|y_i|^{p-1}] > 0 \quad (6.3)$$

Moreover, in order to ensure stability the following conditions must be satisfied

$$p^2 E[y_i^2] E[y_j^2] E[|y_i|^{p-1}] E[|y_j|^{p-1}] > 1. \quad (6.4)$$

Assuming that normalization condition (4.6) is imposed, i.e.

$$E[|y_i|^{p+1}] = 1 \quad (6.5)$$

and introducing a generalized normalized kurtosis (Gray's norm)

$$\kappa_{pi} = \frac{E[|y_i|^{p+1}]}{E[y_i^2] E[|y_i|^{p-1}]} > 0 \quad (6.6)$$

we obtain simple conditions for stability:

$$\kappa_{pi} \kappa_{pj} < p^2. \quad (6.7)$$

From the above inequalities it is seen that they could be satisfied only for  $p > 1$ . For  $p = 1$  (a linear function) the conditions are not satisfied since  $\kappa_{1i} = 1$ . In the special case of cubic function  $\varphi_i(y_i) = y_i^3$  the conditions are satisfied, for example, if all signals are sub-Gaussian, i.e. the standard normalized kurtosis:

$$\kappa_{4i} = \frac{E[|y_i|^4]}{E^2[y_i^2]} < 3. \quad (6.8)$$

This results have been fully confirmed by our computer simulation experiments.

#### Example 2

In the second example we consider a symmetrical sigmoidal odd function:

$$\varphi_i(y_i) = \tanh(\gamma y_i). \quad (6.9)$$

Again it is easy to check that conditions (5.8) and (5.9) are satisfied for any positive  $\gamma$ . The condition (5.10) can be expressed as

$$\gamma^2 E[y_i^2] E[y_j^2] E[1 - (\gamma y_i)^2 + \frac{2}{3}(\gamma y_i)^4] E[1 - (\gamma y_j)^2 + \frac{2}{3}(\gamma y_j)^4] > 1 \quad (6.10)$$

assuming the following approximation of the nonlinear function:

$$\varphi_i(y_i) = \varphi_i(y_i) \approx \gamma y_i - \frac{1}{3}(\gamma y_i)^3 + \frac{2}{15}(\gamma y_i)^5. \quad (6.11)$$

Let us assume now for the simplicity that all output signals are normalized as

$$E[(\gamma y_i)^2] = 1, \quad (6.12)$$

then the stability conditions could be expressed in a simple form as

$$\kappa_{4i}\kappa_{4j} > \frac{9\gamma^2}{4}. \quad (6.13)$$

It is easy to check that these conditions are satisfied, e.g. for  $\gamma < 2$  and for super-Gaussian signals for which the normalized standard kurtosis  $\kappa_{4i} = E[y_i^4]/E^2[y_i^2] > 3$ .

However, our stability conditions are much weaker and in fact the signals do not need necessary to be super-Gaussian.

## 7 Universally convergent learning algorithm

When the stability conditions are not satisfied, the separating solution  $\mathbf{W} = \mathbf{A}^{-1}$  is unstable and we cannot reach it by the learning algorithm. This occurs typically when both super Gaussian and sub-Gaussian source signals exist. If we can observe that conditions (5.8) and (5.9) hold (in fact, they are always satisfied for continuous monotonic increasing functions) but (5.10) do not hold, there exists a simple way to stabilize it.

Let us assume that (5.10) does not hold for a specific pair  $(i, j)$ . In terms of  $dx_{ij}$ , this part of the original learning equation is

$$\begin{pmatrix} \dot{x}_{ij} \\ \dot{x}_{ji} \end{pmatrix} = -\mu(t) \begin{pmatrix} \varphi_i(y_i)y_j \\ \varphi_j(y_j)y_i \end{pmatrix}. \quad (7.1)$$

Its Hessian

$$\mathbf{Q}_{ij} = \begin{bmatrix} \sigma_j^2 k_i & 1 \\ 1 & \sigma_i^2 k_j \end{bmatrix} \quad (7.2)$$

is negative-definite. Let us define a block matrix

$$\mathbf{C}_{ij} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad (7.3)$$

where  $\mathbf{C}_{ij}^2 = \mathbf{I}$ . Then, we have

$$\mathbf{Q}_{ij}\mathbf{C}_{ij}\mathbf{C}_{ij} \begin{bmatrix} \varphi_i(y_i)y_j \\ \varphi_j(y_j)y_i \end{bmatrix} = (\mathbf{Q}_{ij}\mathbf{C}_{ij}) \begin{bmatrix} \varphi_j(y_j)y_i \\ \varphi_i(y_i)y_j \end{bmatrix}. \quad (7.4)$$

The 2 by 2 matrix  $\mathbf{Q}_{ij}\mathbf{C}_{ij}$  is not symmetric. But it is easy to show that the real parts of its eigenvalues are positive when (5.10) does not hold, except the special case when  $\sigma_i^2\sigma_j^2k_ik_j = 1$ . Hence, let  $\mathbf{C}_{ij}$  be the operator which interchanges  $\varphi_j(y_j)y_i$  with  $\varphi_i(y_i)y_j$  and  $\tilde{\mathbf{C}} = \prod \mathbf{C}_{ij}$  for which (5.10) does not hold. Put

$$\dot{\mathbf{W}}(t) = \mu(t) \prod \mathbf{C}_{ij} \circ (I - \varphi(\mathbf{y})^T \mathbf{y}) \mathbf{W}(t). \quad (7.5)$$

This simple modification does not change the equilibrium  $\mathbf{W} = \mathbf{A}^{-1}$ , but its Hessian  $\mathbf{Q}$  is now positive-definite. The operator  $\mathbf{C}_{ij}$  should be taken for all the pairs for which (5.10) does not hold. **Theorem 2** The separating solution is always a stable equilibrium of the modified learning equation (7.5) except for the case of  $\sigma_i^2\sigma_j^2k_ik_j = 1$ .

The quadratic form  $d^2l$  is diagonalized two-by-two block-wise. More specifically,  $(dx_{ii})^2$  parts are diagonalized and  $(dx_{ij}, dx_{ji})$  parts are two-by-two diagonal with diagonal parts equal to  $\mathbf{C}_{ij}\mathbf{Q}_{ij}$ . This fact was first given by Cardoso and Laheld (1996) in a special case.

Hence, we can drive a more general universally converging algorithm by using the inverse of the Hessians, provided that  $\sigma_i^2$ ,  $m_i$  and  $k_i$  are estimated adaptively. Then, the learning equation

$$\dot{\mathbf{W}}(t) = \mu(t)\mathbf{Q}^{-1} \circ [\mathbf{I} - \varphi(\mathbf{y}(t))^T \mathbf{y}(t)] \mathbf{W}(t) = \mu(t)\mathbf{G}[\mathbf{y}(t)]\mathbf{W}(t), \quad (7.6)$$

where  $\mathbf{Q}$  is the Hessian operator consisting of the diagonal blocks  $\mathbf{Q}_{ij}$  and components (entries) of matrix  $\mathbf{G}[\mathbf{y}(t)]$  are expressed as

$$g_{ij} = \frac{1}{\sigma_i^2\sigma_j^2k_ik_j - 1} \left\{ \sigma_j^2k_i\varphi(y_i)y_j - \varphi(y_j)y_i \right\}, \quad i \neq j \quad (7.7)$$

$$g_{ii} = \frac{1}{m_i + 1} \{1 - \varphi(y_i)y_i\}, \quad (7.8)$$

The above results can be summarized by the following Theorem.

**Theorem 3.** The separating solution is always a stable equilibrium of the modified learning equation (7.6).

It should be noted that the Hessian of this new learning equation is the identity. This implies that the convergence is isotropic, that is, it converges equally well for any direction.

## 8 Conclusions

In this paper we derived a family of adaptive learning algorithms with equivariant property for both feed-forward and recurrent neural network models. Necessary and sufficient conditions for their local stability are derived and proved. A new universal algorithm has been proposed which provides stability for any distribution of signals. Illustrative examples demonstrate usefulness of the proposed approach.

## References

- [1] S. Amari and J.F. Cardoso (1996), Blind source separation — Semi-parametric statistical approach, submitted (1996).
- [2] S. Amari (1967), A theory of adaptive pattern classifiers, *IEEE Trans. on Electronic Computers*, vol. EC-16, No.3, 1967, pp.299-307.
- [3] S. Amari (1997), Natural gradient works efficiently in learning, *Neural Computation* (submitted).
- [4] S. Amari, A.Cichocki and H. Yang (1996), A new learning algorithm for blind signal separation, *Advances in Neural Information Processing Systems (1995)*, **8**, eds David S. Touretzky, Michakel C. Mozer and Michael E. Hasselmo, MIT Press: Cambridge, MA, pp. 757-763, 1996.
- [5] S. Amari, A. Cichocki and H. Yang (1995), Recurrent neural networks for blind separation of sources, *Proceedings of Int. Symposium on Nonlinear Theory and its Applications, NOLTA-95*, Las Vegas, December 1995, 37–42.
- [6] S. Amari and M. Kawanabe (1996), Information geometry of estimating functions in semiparametric statistical models, *Bernoulli*, **2**(3).
- [7] S. Amari and M. Kawanabe (1996b), Estimating functions in semiparametric statistical models, *Estimating Functions*, edited by V.P. Godambe (in press)

- [8] A. J. Bell and T. J. Sejnowski (1995), An information-maximization approach to blind separation and blind deconvolution, *Neural Computation*, **7**, pp. 1129-1159.
- [9] P. J. Bickel, C. A. J. Klaassen, Y. Ritov and J. A. Wellner (1993), *Efficient and Adaptive Estimation for Semiparametric Models*, Baltimore, MD: Johns Hopkins University Press.
- [10] J.F. Cardoso and B. Laheld (1996), Equivariant adaptive source separation, *IEEE Trans. on Signal Processing*, Dec. 1996 (in print).
- [11] P. Common (1994), Independent component analysis, a new concept?, *Signal Processing*, **36**, pp.287–314.
- [12] T. Chen and R. Chen (1994), Blind extraction of stochastic and deterministic signals by neural network approach, 28th Asilomar Conference on Signals, Systems and Computer, Pacific Gr., CA, USA, IEEE Computer Society, pp.892-896.
- [13] A. Cichocki, R. Unbehauen and E. Rummert (1994), Robust learning algorithm for blind separation of signals, *Electronics Letters*, vol. 30, no. 17, August 1994, 1386–1387.
- [14] A. Cichocki, R. Unbehauen, L. Moszczyński and E. Rummert, A new on-line adaptive learning algorithm for blind separation of source signals, in *Proc. of ISANN-94*, Taiwan, 1994, 406–411.
- [15] A. Cichocki, S. Amari, M. Adachi, and W. Kasprzak (1996), Self-adaptive neural networks for blind separation of sources, *Proceedings of Int. Symp. on Circuits and Systems (ISCAS-96)*, May 1996, Atlanta, GA, USA, vol. 2, 157–161.
- [16] Y. Deville (1996), A unified stability analysis of the Herault-Jutten source separation neural network, *Signal Processing*, vol.51, No.3, 1996, pp.229-233.
- [17] J.C. Fort (1991), Stabilité de l'algorithme de séparation de sources de Jutten et Herault, *Traitement du Signal*, vol. 8 no.1, pp. 35-42.

- [18] J. Herault and C. Jutten (1986), Space or time adaptive signal processing by neural network models, In: J. S. Denker (ed.): *Neural Networks for Computing. Proceedings of AIP Conference*. American Institute of Physics, New York, 1986, 206–211.
- [19] C. Jutten and J. Herault (1991), Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture, *Signal Processing*, vol. 24, 1991, 1–20.
- [20] O. Macchi, E. Moreau (1997), Self-adaptive source separation Part I: convergence analysis of a direct linear network controlled by Herault-Jutten algorithm. Submitted to IEEE Transactions on Signal Processing.
- [21] E. Oja and J. Karhunen (1995), Signal separation by nonlinear Hebbian learning, in M. Palaniswami et al. (Eds.), *Computational Intelligence - A Dynamic System Perspective*, New York, IEEE Press, 1995, 83–97.
- [22] E. Sorouchyari (1991), Blind separation of sources, part III: stability analysis, *Signal Processing*, Vol. 24, pp.21-30.