A Pair-Channel Learning Algorithm with Constraints for Multi-Channel Blind Separation

Kenji NAKAYAMA Akihiro HIRANO Takayuki SAKAI

Dept of Information and Systems Eng., Faculty of Eng., Kanazawa Univ.
2–40–20 Kodatsuno, Kanazawa, 920–8667, Japan
e-mail: nakayama@t.kanazawa-u.ac.jp

Abstract

A pair learning algorithm, based on Jutten's algorithm, is proposed in this paper. A feedback weight $c_{ij}(n)$ from the jth output $y_j(n)$ to the ith observation $x_i(n)$ is updated using only the ith and jth outputs. Then, this method is called "Pair Learning Algorithm". This algorithm is compared with a global learning algorithm, which use all outputs in updating the feedback weights. The number of computations is reduced and a learning process is simplified. Separation performance by both methods are not so different. Furthermore, the following properties are analyzed: Order of the separated signal sources is dependent on the power ratio of the signal sources included in the observations. Magnitude of the separated signal sources are uniquely determined in the pair learning algorithm after convergence. Squashing nonlinear functions have effects of stabilizing the learning process. Simulation using voices, white noise and music signal are demonstrated.

1 Introduction

Signal processing including noise cancelation, echo cancelation, equalization of transmission lines, estimation and restoration of signals have been becoming very important technology. In some cases, we do not have enough information about signals and interference. Furthermore, their mixing process and transmission processes are not well known in advance. Under these situations, blind source separation methods using statistical property of the signal sources have become important [1]-[5].

Jutten et all proposed a blind separation algorithm based on statistical independence and symmetrical distribution of the signal sources [6]-[8]. The learning algorithm is derived assuming approximate convergence. That is, some of the outputs are expected to be similar to the signal sources. When some signal source level is dominant, it is difficult to make the outputs statistically independent. Under these situations, Jutten's learning rule does not work well.

In order to overcome this problem, two stabilization methods have been proposed [10]. Unstable behavior caused by corruption of symmetrical distribution and imbalance of the signal source levels can be overcome.

In this paper, a pair learning algorithm is proposed for multi-channel blind source separation. Based on the Jutten's BSS method, a feedback weight c_{ji} from the jth output to the ith observation is updated using only the ith and jth outputs $y_i(n)$, $y_j(n)$. Then, this method is called "Pair Learning Algorithm". On the other hand, c_{ji} is updated using $y_i(n)$ and $y_j(n), j = 1, 2, \cdots, N$ in [12]. These learning methods are compared based on convergence properties, order and magnitude of the separated signals. Furthermore, effects of nonlinear functions on convergence is investigated. Computer simulation will be demonstrated using noise, voices and music signal sources.

2 Jutten's Blind Separation

2.1 Network Structure

Figure 1 shows a blind separation model proposed by Jutten et all [6].

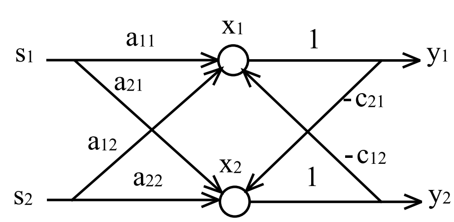

Figure 1: Block diagram of Jutten's blind separation.

The number of the signal sources, the sensors and the outputs are all the same. The signal sources $s_i(n), i = 1, 2, \cdots, N$ are linearly combined using unknown weights a_{ji}, and are sensed at N points, resulting in $x_j(n)$.

$$x_j(n) = \sum_{i=1}^{N} a_{ji} s_i(n) \tag{1}$$

The output of the separation block $y_k(n)$ is given by

$$y_j(n) = x_j(n) - \sum_{\substack{k=1 \\ \neq j}}^{N} c_{jk} y_k(n) \tag{2}$$

This relation is expressed using vectors and matrices in the case of $N = 2$ as follows:

$$\boldsymbol{x}(n) = \boldsymbol{A}\boldsymbol{s}(n) \tag{3}$$

$$\boldsymbol{y}(n) = \boldsymbol{x}(n) - \boldsymbol{C}\boldsymbol{y}(n) \tag{4}$$

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \boldsymbol{C} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \tag{5}$$

$\boldsymbol{A}$ is an unknown mixing matrix. From these expressions, a relation between the signal sources and the separation outputs becomes

$$\boldsymbol{y}(n) = (\boldsymbol{I} + \boldsymbol{C})^{-1}\boldsymbol{x}(n) = (\boldsymbol{I} + \boldsymbol{C})^{-1}\boldsymbol{A}\boldsymbol{s}(n) \tag{6}$$

The following matrix can be regarded as a separation matrix.

$$\boldsymbol{W} = (\boldsymbol{I} + \boldsymbol{C})^{-1} \tag{7}$$

In order to evaluate separation performance, the following matrix is defined.

$$\boldsymbol{P} = \boldsymbol{W}\boldsymbol{A} \tag{8}$$

If $\boldsymbol{P}$ takes the next forms, the signals s_1 and s_2 are completely separated at ether y_1 or y_2.

$$\boldsymbol{P} = \begin{bmatrix} p_{11} & 0 \\ 0 & p_{22} \end{bmatrix} \quad or \quad \begin{bmatrix} 0 & p_{12} \\ p_{21} & 0 \end{bmatrix} \tag{9}$$

2.2 Learning Algorithm

Error Function

It is assumed that the separation block approaches the optimum solution, then $N-1$ outputs are already proportional to the signal sources. Only one output has not reached to the signal source. Let $y_l(n)$ be this output. From Eq.(2),

$$y_l(n) = x_l(n) - \sum_{\substack{k=1 \\ \neq l}}^{N} c_{lk} y_k(n) \tag{10}$$

Here, y_k is already $a_{kk}s_k(n)$. Furthermore, $x_l(n)$ is replaced by Eq.(1).

$$y_l(n) = \sum_{\substack{k=1 \\ \neq l}}^{N} (a_{lk} - c_{lk}a_{kk}) s_k(n) + a_{ll} s_l(n) \tag{11}$$

The mean squared $y_l(n)$ becomes

$$\begin{aligned} J_l &= E[y_l^2(n)] \\ &= \sum_{\substack{k=1 \\ \neq l}}^{N} (a_{lk} - c_{lk}a_{kk})^2 E[s_k^2(n)] + a_{ll}^2 E[s_l^2(n)] \end{aligned} \tag{12}$$

In the above derivation, $s_k(n)$ are assumed to be independent to each other. Furthermore, it is assumed that mean of $s_i(n)$ is zero, then mean of $y_k(n)$ are also zero.

In $E[y_l^2(n)]$, only $(a_{lk} - c_{lk}a_{kk})^2$ can be controlled by the separation block. It can be zero for $c_{lk} = a_{lk}/a_{kk}$. Thus, the optimum solution can be obtained employing J_l as a cost function.

Assumptions of Signal Source Properties

Assumption 1: The signal sources are statistically independent to each other.

Assumption 2: Samples of the signal sources are symmetrically distributed, that is their probability density function (pdf) are even.

The update equation is given by

$$c_{lk}(n+1) = c_{lk}(n) + \eta f(y_l(n)) g(y_k(n)) \tag{13}$$

$f()$ and $g()$ are odd nonlinear functions.

3 Stabilization of Learning Process

3.1 Even Probability Density Function

Reason for Unstable Behavior

Jutten's algorithm assume the even pdf for $s_i(n)$, furthermore, $y_k(n)$ are close to $s_i(n)$. However, during early stage of the learning process, these assumptions are not guaranteed. Although the even pdf can be assumed, it is not valid for a small number of samples. Large samples easily disturb the even pdf condition.

Stabilization: Method 1

In order to avoid disturbance by large samples, they are detected and the learning of $c_{lk}(n)$ is skipped. First, the variance $\sigma_{y_k}^2(n)$ is estimated using the output samples. If one of $|y_k(n)|$ exceeds $\theta_L \sigma_{y_k}(n)$, the learning is skipped [10].

3.2 Imbalance of Signal Source Powers

Silent Intervals and Power Imbalance

When the signal sources are voices, silent intervals are included. In this interval, the outputs $y_k(n)$ have strong correlation. The adaptation in this interval causes divergence. This problem will occurs not only in the silent intervals but also in signal power imbalance. The cross-correlation between the observations is used for detecting signal power imbalance.

Stabilization: Method 2

Using the cross-correlation $\rho_{x_1 x_2}$, the intervals, where the signal powers are imbalance, are detected. If $\rho_{x_1 x_2} > \theta_C$, the learning process is skipped [10].

4 Multi-Channel Blind Source Separation

The network of 3-channels is shown in Fig.2.

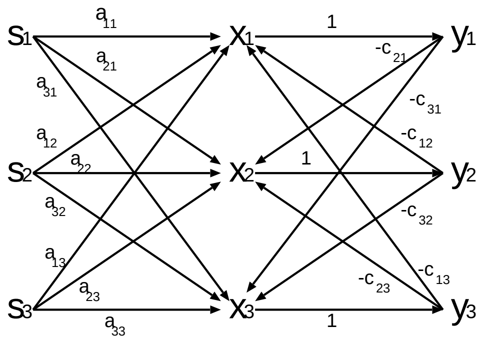

Figure 2: Block diagram for 3 signal sources, 3 sensors and 3 separations (3-3-3 model).

4.1 Pair Learning Algorithm

As shown in Eq.(13), $c_{lk}(n)$ is updated using the outputs $y_l(n)$ and $y_k(n)$. This updating process is applied to the multi-channel environment. This process can be expressed by

$$\begin{aligned} \boldsymbol{C}(n+1) &= \boldsymbol{C}(n) + \eta[f(\boldsymbol{y}(n))g(\boldsymbol{y}^T(n)) \\ &- \boldsymbol{\Lambda(n)}] \quad (14) \\ \boldsymbol{C}(n) &= \{c_{ij}\}, \quad c_{ii}(n) = 0 \quad (15) \\ \boldsymbol{y}(n) &= [y_1(n), y_2(n), \cdots, y_N(n)]^T \quad (16) \\ f(\boldsymbol{y}) &= [f_1(y_1(n)), \cdots, f_N(y_N(n))]^T \quad (17) \\ g(\boldsymbol{y}) &= [g_1(y_1(n)), \cdots, g_N(y_N(n))]^T \quad (18) \\ \boldsymbol{\Lambda}(n) &= diag[f_1(y_1(n))g_1(y_1(n), \\ & \cdots, f_N(y_N(n))g_N(y_N(n))] \quad (19) \end{aligned}$$

The number of the feedback connections is $N(N-1)$. The number of multiplications used in updating the $N(N-1)$ weights is a sum of $N(N-1)$ for $f_i(y_i(n))g_j(y_j(n))$ and N for $\eta f_i(y_i(n))$, then totally N^2 multiplications are required.

4.2 Gloval Learning Algorithm

The update processes for a feedforward network and a fully recurrent network are given by [11],[12]

Feedforward Network

$$\begin{aligned} \boldsymbol{W}(n+1) &= \boldsymbol{W}(n) + \eta(n)[\boldsymbol{\Lambda}(n) \\ &- \phi(\boldsymbol{y}(n))\boldsymbol{y}^T(n)]\boldsymbol{W}(n) \quad (20) \end{aligned}$$

Fully Recurrent Network

$$\begin{aligned} \boldsymbol{C}(n+1) &= \boldsymbol{C}(n) + \eta(n)[\boldsymbol{C}(n) + \boldsymbol{I}][\boldsymbol{\Lambda}(n) \\ &- \phi(\boldsymbol{y}(n))\boldsymbol{y}^T(n)] \quad (21) \end{aligned}$$

In Eq.(21), for instance, $c_{ik}(n)$ is updated as,

$$\begin{aligned} c_{ik}(n+1) &= c_{ik}(n) + \eta \sum_{j=1}^{N} c'_{ij}(n) b_{jk}(n) \quad (22) \\ c'_{ii}(n) &= 1 \\ b_{jk}(n) &= -\phi_j(y_j(n))y_k(n) \\ b_{jj}(n) &= 0 \quad (23) \end{aligned}$$

$\boldsymbol{\Lambda}(n)$ can be replaced by $\boldsymbol{I}$. $c_{ik}(n)$ is updated using overall output information, that is $y_k(n)$ and $y_j(n), j = 1, 2, \cdots, N, \neq k$. However, it has some constraints. $\phi_j(y_j(n))y_k(n)$ is weighted with $c_{ij}(n)$. This weighting is the same for the elements in the ith row of $\boldsymbol{C}(n+1)$. The number of the feedback connections is also $N(N-1)$. $N(N-1)$ multiplications are required for $\boldsymbol{Y}(n) = \boldsymbol{\Lambda}(n) - \phi(\boldsymbol{y}(n))\boldsymbol{y}^T(n)$. Furthermore, N^3 multiplications are required for $[\boldsymbol{C}(n) + \boldsymbol{I}]\boldsymbol{Y}(n)$. However, N^3 can be reduced to $O(N^2)$ by changing the order of calculations. Still, this algorithm requires more computations compared from the pair learning algorithm, while more information are used for updating the connection weights.

5 Some Considerations

5.1 Nonlinear Functions

The nonlinear functions, $f_i(y_i(n))$, $g_i(y_i(n))$, $\phi_i(y_i(n))$, are odd functions due to the assumption of the even pdf for the signal sources. Still, there are many selections for the odd nonlinear functions. In this paper, squashing and non-squashing functions are compared. As described in Sec.3, the assumption of the even pdf is easily broken by large samples in an early learning stage. Squashing functions compress large amplitude samples, so stabilization effects can be expected. Also, a combination of the nonlinear functions must be optimized.

5.2 Order of Separated Signal Sources

Generally speaking, there is no information to determine the order of the separated signal sources. The order means the output terminals, where the signal sources are separated. However, it is highly dependent on dominant components in the outputs in the early learning stage. In the pair learning algorithm, based on Jutten's algorithm, if the feedback weights are initially zero, then the output $y_j(n)$ are equal to the observation $x_j(n)$. Letting $x_j(n)$ be a linear combination of the signal sources, they are related as

$$x_j(n) = \sum_{i=1}^{N} a_{ji} s_i(n) \tag{24}$$

$$y_j(n) = x_j(n) \tag{25}$$

If the signal sources $s_{i1}(n)$ and $s_{i2}(n)$ are dominant in $y_{j1}(n)$ and $y_{j2}(n)$, respectively, the correction terms for the feedback weights $c_{j1,j2}$ and $c_{j2,j1}$ are mainly determined by the dominant components. They are adjusted so as to extract $s_{i1}(n)$ and $s_{i2}(n)$ in $y_{j1}(n)$ and $y_{j2}(n)$, respectively. In other words, $s_{i1}(n)$ and $s_{i2}(n)$ in $y_{j1}(n)$ and $y_{j2}(n)$ suppress the same components in the other outputs, like competitive learning. Thus, the signal sources will remain at the outputs where they occupy dominant at the beginning of the learning process.

The global learning algorithms have the same property as in Eq.(25), if the connection weight matrix $\boldsymbol{W}(n)$ is initially set to the identity matrix $\boldsymbol{I}$ in the forward type, and $c_{jk}(0) = 0$ in the fully recurrent type.

5.3 Silence of Signal Sources

Another aspect regarding the order of the separated signal sources is silence of the signal sources. When some of the signal sources are silence in some intervals, what happen in the order of separation? Suppose $s_i(n)$ is separated at $y_k(n)$, and $s_i(n)$ becomes zero in some interval, which signal source is separated at $y_k(n)$? Since $s_i(n)$ is separated at $y_k(n)$, it can be expressed by

$$y_k(n) = s_i(n) + \bar{s_i}(n) \tag{26}$$

$\bar{s_i}(n)$ is the cross term of the other signal sources, which is well reduced after convergence. For instance, their ratio is about 20∼30 dB in practical applications. Thus, when $s_i(n) = 0$, $y_k(n)$ become small value, and the correction term given by $f_k(y_k(n))g_l(y_l))$ and $f_l(y_l(n))g_k(y_k(n))$ become also small values. This means updating the weights $c_{kl}(n)$ and $c_{lk}(n)$ are very slow. $y_k(n)$ can stay at a small value for a while. Of course, if $s_i(n)$ is silent for a very long period, another signal source will appear. However, this situation is easily repaired when $s_i(n)$ comes back, that is $s_i(n)$ can easily occupy $y_k(n)$.

In the pair learning algorithm, the same signal source can be separated at the different outputs. Let $s_{i1}(n)$ be separated at $y_{k1}(n)$, and consider $s_{i1}(n)$ is also separated at $y_{k2}(n)$. The feedback weights $c_{k1,k2}(n)$ and $c_{k2,k1}(n)$ are not updated due to the stabilization using the cross-correlation between $x_{k1}(n)$ and $x_{k2}(n)$, in which $s_{i1}(n)$ is dominant. However, the other feedback weights $c_{k1,x}$ and $c_{x,k1}$ are updated, where $x \neq k2$.

5.4 Magnitude of Separated Signal Sources

There is no information about power of the signal sources. After convergence, the connection weights satisfy the following situations. Let $s_{i1}(n)$ be separated at $y_{k1}(n)$, then $s_{i1}(n)$ appears at the observation $x_j(n)$ are cancelled by $s_{i1}(n)$ in $y_{k1}(n)$. This cancellation is expressed as,

$$c_{j,k1}(n) a_{k1,i1} = a_{j,i1} \tag{27}$$

$$c_{j,k1}(n) = \frac{a_{j,i1}}{a_{k1,i1}} \tag{28}$$

As described in Sec.2.2, the cost function of Jutten's algorithm can guarantee the above feedback weights after successful convergence. Therefore, even though we cannot estimate in advance, magnitude of the separated signal sources are uniquely determined as,

$$\text{Separated} \;\; s_{i1}(n) = a_{j1,i1} s_{i1}(n) \tag{29}$$

On the other hand, in the global learning methods, there is no constraint on magnitude of separated signals [11],[12].

6 Simulation and Discussions

6.1 Simulation Conditions

Nonlinear Functions

Case-1

$$f_1(y) = y^3, \quad g_1(y) = y \tag{30}$$

$$\phi_1(y) = y^3 \tag{31}$$

Case-2

$$f_2(y) = \frac{1-e^{-5y}}{1+e^{-5y}}, \quad g_2(y) = \frac{1-e^{-y}}{1+e^{-y}} \tag{32}$$

$$\phi_2(y) = \frac{1-e^{-5y}}{1+e^{-5y}} \tag{33}$$

Learning Rate

The learning rate is $\eta = 0.005$ for SNR$\leq$ 20dB, and 0.001 for SNR$> 20dB$ in both methods. SNR is defined by using $\boldsymbol{P}$ in Eq.(8)

$$\text{SNR} = 10 \log \frac{\sum_{i,j\in\Omega_1} p_{ij}^2}{\sum_{i,j\in\Omega_2} p_{ij}^2} \tag{34}$$

p_{ij} are elements of $\boldsymbol{P}$. Ω_1 includes the elements of the separated signal sources, and Ω_2 includes the elements of the cross terms.

Mixing Matrix
SNR is always 2 dB for both methods and all cases of the number of channels. The definition of SNR for the mixing matrices is the same as in Eq.(34).

$$\boldsymbol{A}_{2ch} = \begin{bmatrix} 1.0 & 0.5 \\ 0.5 & 1.0 \end{bmatrix} \tag{35}$$

$$\boldsymbol{A}_{3ch} = \begin{bmatrix} 1.0 & 0.7 & 0.3 \\ 0.4 & 1.0 & 0.5 \\ 0.8 & 0.5 & 1.0 \end{bmatrix} \tag{36}$$

$$\boldsymbol{A}_{4ch} = \begin{bmatrix} 1.0 & 0.3 & 0.5 & 0.3 \\ 0.5 & 1.0 & 0.6 & 0.3 \\ 0.4 & 0.5 & 1.0 & 0.4 \\ 0.5 & 0.5 & 0.3 & 1.0 \end{bmatrix} \tag{37}$$

$$\boldsymbol{A}_{5ch} = \begin{bmatrix} 1.0 & 0.3 & 0.4 & 0.3 & 0.5 \\ 0.4 & 1.0 & 0.2 & 0.5 & 0.3 \\ 0.3 & 0.4 & 1.0 & 0.3 & 0.3 \\ 0.3 & 0.4 & 0.6 & 1.0 & 0.3 \\ 0.3 & 0.5 & 0.3 & 0.3 & 1.0 \end{bmatrix} \tag{38}$$

Signal Sources
2-channels: music
3-channels: two voices and one white noise
4-channels: three voices and one white noise
5-channels: four voices and one white noise

6.2 Separation Performances

Voices and White Noise Sources

Figure 3 shows the learning curves of the pair learning algorithm using the Case-2 nonlinear functions and the stabilization with the cross-correlation. The vertical and horizontal axes indicate SNR dB and the number of iterations, respectively. Since the squashing functions are used, the stabilization for large output samples is not necessary. Figure 4 shows the results without the stabilization based on the cross-correlation among the observations. The results are almost the same for 3-channel case. However, in the cases of 4- and 5-channels, the cross-correlation stabilization is effective. The difference is not so much. From these results, it is confirmed that the squashing nonlinear functions are useful for stabilizing the learning process.

Figure 5 shows the learning curves for the global learning algorithm using the Case-2 nonlinear function. No stabilization is required. This approach can provide higher SNR for 4-channel case, while the results in 3- and 5-channels are almost the same as in Figs.3 and 4.

2-Channels and Music Signal Sources

Figures 6 and 7 compare the pair and the global algorithms using 2-channel music signal sources. The Case-1 and Case-2 nonlinear functions are used in Figs.6 and

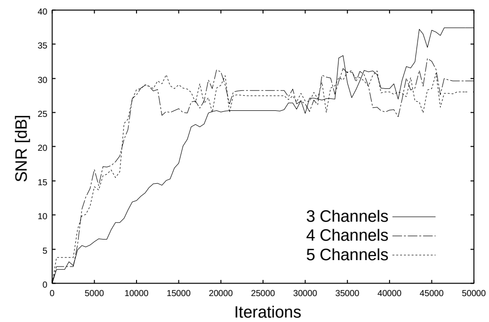

Figure 3: Learning curves of pair learning using Case-2 nonlinear functions and stabilization with cross-correration.

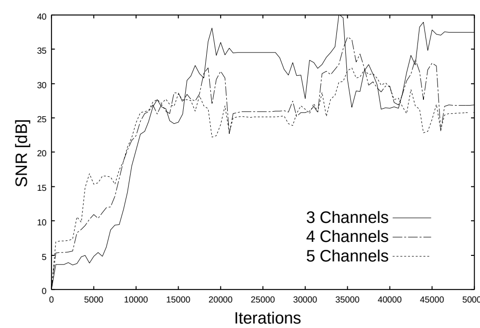

Figure 4: Learning curves of pair learning using Case-2 nonlinear functions. Stabilization with cross-correlation is not used.

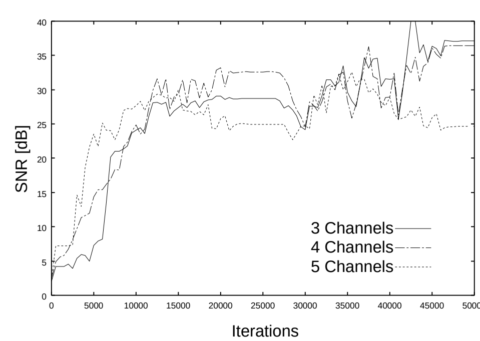

Figure 5: Learning curves of global learning using Case-2 nonlinear functions.

7, respectively. In Fig.6, both the stabilization methods described in Sec.3 are used. Since the number of channels is two, there is no essential difference between the pair and the global learning algorithms. The difference is rather caused by the nonlinear functions and the stabilization methods. As a result, a combination of non-squashing functions and the stabilization methods is useful for the music signal sources. However, this point should be more investigated.

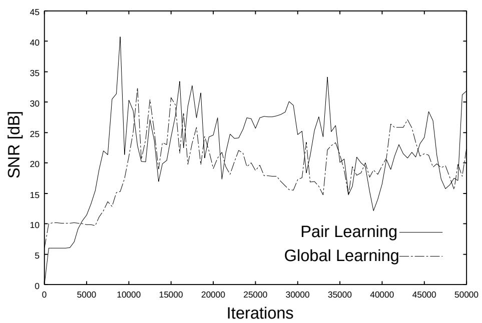

Figure 6: Learning curves of pair and global learning algorithms using Case-1 nonlinear functions. 2-channel music signal sources are used.

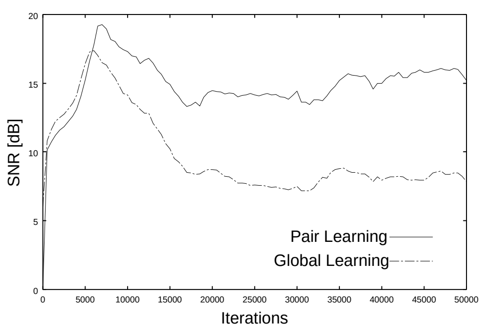

Figure 7: Learning curves of pair and global learning algorithms using Case-2 nonlinear functions. 2-channel music signal sources are used.

7 Conclutions

For multi-channel blind signal source separation, a pair learning algorithm has been proposed. This algorithm is a simplified version of the global learning algorithm. The number of computations can be reduced, and a learning process is also simplified. Separation performances are almost the same as the global methods. Furthermore, nonlinear functions are compared using squashing and non-squashing functions. The squashing nonlinear functions have effects of stabilizing the learning process. However, efficiency is dependent on nature of signal sources. For instance, in the case of music signal sources, a combination of non-squashing functions and the stabilization methods is useful. This point should be more investigated.

References

[1] C.Jutten, J.Herault and A.Guerin, "IIN.C.A: An independent components analyzer based on an adaptive neuromimetic network", in: J.Degmongeot, T.Herve, V.Raille and C.Roche, eds., Artificial Intelligence and Cognitive Science, Manhester Univ. Press, Manchester, 1988.

[2] C.Jutten and J.Herault, "Analog implementation of permanent unsupervised learning algorithm", Proc. NATO Advanced Research Workshop on Neurocomputing, Les Arcs, France, pp.145-152, Feb.27-March 3, 1989.

[3] P.Comon, "Separation of stochastic process whose linear mixtures observed", Proc. ONR- NSF-IEEE Workshop on Higher Spectral Analysis Vail, Colorado, pp.174-179, June 28-30, 1989.

[4] P.Comon,"Separation of sources using higher-order cumulants", SPIE Conference, Vol.1152, Advanced Algorithms and Architectures for Signal Processing IV, San Diego, California, pp. 170-181, August 6-11, 1989.

[5] J.F.Cardoso, "Eigen structure of the 4th order cumulant tensor with application to the blind source separation problem", ICASSP Proc. pp. 2655-1658.

[6] C.Jutten and Jeanny Herault, "Blind separation of sources, Part I: An adaptive algorithm based on neuromimetic architecture", Signal Processing, 24, pp.1-10, 1991.

[7] P.Comon, C.Jutten and J.Herault, "Blind separation of sources, Part II: Problems statement", Signal Processing, 24, pp.11-20, 1991.

[8] E.Sorouchyari, "Blind separation of sources, Part III: Stability analysis", Signal Processing, 24, pp.21-29, 1991.

[9] S.Haykin, Adaptive Filter Theory, 3rd ed., Prentice-Hall, Inc. 1996.

[10] K.Nakayama, A.Hirano and M.Nitta, "A constraint learning algorithm for blind source separation", IEEE INNS, Proc. IJCNN'2000, Como, Italy, pp.24-27, July, 2000.

[11] A.Cichocki, S.Amari, M.Adachi, W.Kasprzak, "Self-adaptive neural networks for blind separation of sources", Proc. ISCAS'96, Atlanta, pp.157-161, 1996.

[12] S.Amari, T.Chen and A.Cichocki,"Srability analysis of learning algorithms for blind source separation", Neural Networks, vol.10, no.8, pp.1345-1351, 1997.