Extraction of Specific Signals with Temporal Structure

Allan Kardec Barros
Bio-mimetic Control Research Center, RIKEN, Moriyama-ku, Shimoshidami, Nagoya 463-0003, Japan

Andrzej Cichocki
Brain Science Institute, RIKEN, Wako-shi, Saitama 351-01, Japan

In this work we develop a very simple batch learning algorithm for semi-blind extraction of a desired source signal with temporal structure from linear mixtures. Although we use the concept of sequential blind extraction of sources and independent component analysis, we do not carry out the extraction in a completely blind manner; neither do we assume that sources are statistically independent. In fact, we show that the a priori information about the autocorrelation function of primary sources can be used to extract the desired signals (sources of interest) from their linear mixtures. Extensive computer simulations and real data application experiments confirm the validity and high performance of the proposed algorithm.

1 Introduction

There has been increased interest in blind signal processing, with special attention to independent component analysis (ICA) and instantaneous blind source separation (BSS) (Amari & Cichoki, 1998; Jutten & Hérault, 1988). Indeed, BSS has been applied to a number of areas, including biomedical signal analysis and processing, geophysical data processing, data mining, speech analysis, image recognition and enhancement, and wireless communications.

BSS is based on the following principle. Assuming that the original (or source) signals have been linearly mixed and that these mixed sensor signals are available, BSS finds in a blind manner a linear combination of the mixed signals that recovers the original source signals, possibly rescaled and randomly arranged in the outputs.

However, extracting all the source signals from a large number of sensors, for example, a magnetoencephalographic (MEG) measurement, which may output hundreds of recordings, could take a long time. Thus, it would be important for the user (e.g., a physician) for the algorithm to extract only one or some desired signals (signals of interest) with given characteristics instead of all sources. A similar problem arises in the cocktail party problem,

Neural Computation **13, 1995–2003** (2001)

where one needs to rearrange speech signals extracted in different frequency bands so that they could recover a given speech signal (Ikeda & Murata, 1999; Lee, 1998). This fact implies a semiblind kind of source separation, which we focus here.

Therefore, it is important to develop algorithms that can extract only the desired signal by using some a priori information. Some work has already been carried out with a similar objective (Barros & Ohnishi, 1999; Luo, Hu, Ling, & Liu, 1999; Barros, Vigário, Jousmäki, & Ohnishi, 2000), but none of them guaranteed theoretically the algorithm convergence to the desired source.

Here we propose using sequential signal extraction along with the a priori information about the autocorrelation function of the signal of interest (Cichocki & Barros, 1999; Cichocki, Thawonmas, & Amari, 1997). In other words, we no longer extract the sources blindly. The sequential estimation is important because we can recover only one specific source from the measured sensor signals instead of extracting all at once.

We will prove here that if certain decorrelation conditions are satisfied, the extraction of all sources is theoretically guaranteed. We also show in practice that this is true by simulations and real-world application.

2 New Approach to Signal Extraction

Let us denote the source signal vector as $\mathbf{s}(k) = [s_1(k), \ldots, s_n(k)]$. In the model, the observed vector $\mathbf{x}(k) = [x_1(k), \ldots, x_n(k)]$ results by linearly mixing the source signals. Thus, this mixture can be written as $\mathbf{x}(k) = \mathbf{A}\mathbf{s}(k)$, where $\mathbf{A}$ is an $n \times n$ nonsingular matrix.

We assume here that the source signals have temporal structure and that they have different autocorrelation functions, but they do not necessarily have to be statistically independent (Amari, 1999; Belouchrani, Meraim, Cardoso, & Moulines, 1997). In fact, their mutual decorrelation in a given range is sufficient (but not necessary) to extract them successfully (see the simulation example in section 4). In other words, in this article, we propose a simple batch algorithm that guarantees blind extraction of any source signal s_i that satisfies for the specific time delay τ_i the following relations:

$$\begin{aligned} &E[s_i(k)s_i(k-\tau_i)] \neq 0 \\ &E[s_i(k)s_j(k-\tau_i)] = 0 \; \forall \; i \neq j. \end{aligned} \tag{2.1}$$

Because we want to extract only a desired source signal, we can use a simple processing unit described as $y(k) = \mathbf{w}^T\mathbf{x}(k)$, where $y(k)$ is the output signal (which estimates the specific source signal s_i), k is the sampling number, and $\mathbf{w}$ is the weight vector. For the purpose of developing the algorithm, let us first define the following error,

$$\varepsilon(k) = y(k) - by(k-p), \tag{2.2}$$

where b is a coefficient of a simple FIR filter with single delay z^{-pT_s}, where T_s is the sampling period (assumed to be one).[1]

2.1 Learning Algorithm. The idea is to carry out the constrained minimization of the mean squared error $\xi(\mathbf{w}, \mathbf{b}) = E[\varepsilon^2]$ for two sets of parameters $\mathbf{w}$ and b. After some simple mathematical manipulations, we find,

$$\xi(\mathbf{w}, \mathbf{b}) = \mathbf{w}^T E[\mathbf{x}\mathbf{x}^T]\mathbf{w} - 2bE[y_p \mathbf{w}^T \mathbf{x}] + b^2 E[\mathbf{y}_p^2]. \tag{2.3}$$

This cost function achieves minimum when its gradient reaches zero in relation to $\mathbf{w}$ and b. Thus, taking into account that $y = \mathbf{w}^T\mathbf{x}$, we find,

$$\frac{\partial \xi(\mathbf{w}, \mathbf{b})}{\partial \mathbf{w}} = 2E[\mathbf{x}\mathbf{x}^T]\mathbf{w} - 2bE[y_p\mathbf{x}] + 2b^2 E[\mathbf{x}_p\mathbf{x}_p] = \mathbf{0}, \tag{2.4}$$

$$\frac{\partial \xi(\mathbf{w}, \mathbf{b})}{\partial b} = 2E[y_p y] - 2bE[y_p^2] = 0. \tag{2.5}$$

These equations yields the following updating rule:

$$\mathbf{w} = E[\mathbf{x}\mathbf{x}^T]^{-1} E[y_p\mathbf{x}] \frac{b}{1 + 2b^2}. \tag{2.6}$$

In order to avoid the trivial solution $\mathbf{w} = \mathbf{0}$, we perform normalization of the vector to unit length at each iteration step as $\mathbf{w}_* = \mathbf{w}//||\mathbf{w}||$. With this, the term $b/(1 + 2b^2)$ can be disregarded. Moreover, we can assume without losing generality that the sensor data are prewhitened; thus, $E[\mathbf{x}\mathbf{x}^T] = \mathbf{I}$. With this, equation 2.6 leads to a very simple learning rule:

$$\mathbf{w} = E[\mathbf{x} y_p]. \tag{2.7}$$

2.2 Estimating the Time Delay. One of the problems with practical implementation is that of how to estimate the optimal time delay τ_i. A simple solution is to calculate the autocorrelation $\varsigma(p) = E[x_j(t)x_j(t-p)]$ of sensor signals as a function of the time delays and find the one corresponding to the pick of $\varsigma(p)$ within an specified interval of interest. Section 4 provides an example of how it can be carried out.

3 Avoiding the Permutation Problem

One of the drawbacks of BSS is that one cannot ensure in which order the sources will be estimated. This is due to an inherent problem originally addressed by Comon (1994); that is, for algorithms that estimate all the

[1] We will drop the sampling number k for simplicity and use the index p for the delay, that is, $y_p = y(k-p)$. We shall use it when necessary.

outputs blindly, say, by using an $n \times n$ matrix $\mathbf{W}$, the output vector $\mathbf{y} = \mathbf{Wx} = \mathbf{WAs}$ has an indeterminacy that can be written as $\mathbf{y} = \mathbf{DPs}$, where $\mathbf{D}$ and $\mathbf{P}$ are diagonal and permutation matrices, respectively. This means that the estimated sources might be randomly rearranged and scaled at the output.

By the above reasoning, because we are here trying to estimate only one source at a time, the indeterminacy can be written as $y = \alpha \mathbf{p}^T \mathbf{s}$, where $\mathbf{p}$ is any column of permutation matrix and α is a scaling factor. Therefore, one cannot predict which source will be estimated at first. However, we believe that random permutation would occur only if we try to recover the sources blindly. Indeed, the following theorem shows that the algorithm given in equation 2.7 solves the permutation problem and leads to the extraction of the target source.

Theorem 1. *Let us define performance vector* $\mathbf{c} = \mathbf{A}^T \mathbf{w}_*$, *where* $\mathbf{w}_*$ *is the vector of weights estimated using the proposed algorithm:*

$$\mathbf{w}_* = E[y_{\tau_i} \mathbf{x}], \qquad ||\mathbf{w}_*|| = 1. \tag{3.1}$$

If for that specific time delay τ_i, *relations 2.1 hold, then* $\mathbf{c} = \beta \mathbf{e}_i$, *where* β *is a nonzero scalar and* $\mathbf{e}_i = [e_1, e_2, \ldots, e_n]$ *is a canonical basis vector, that is,* $e_i = \pm 1$ *and* $e_l = 0, \forall l \neq i$.

Proof. Two stochastic processes, say, $s_i(k)$ and $s_l(m)$, are mutually decorrelated if $E[s_i(k)s_h(l)] = 0$ for every k and l with $i \neq h$ (Papoulis, 1991). From this, we find the correlation matrix $\Psi_{i,\tau_i} = E[\mathbf{s}(k)\mathbf{s}^T(k-\tau_i)] = E[\mathbf{s}\mathbf{s}_{\tau_i}^T]$, where $\psi_{ii,\tau_i} \neq 0$ and $\psi_{gh,\tau_i} = 0, \forall \{g,h\} \neq i$. In other words, all the elements of Ψ_{i,τ_i} are null but ψ_{ii,τ_i}.

From equation 3.1 and the fact that $\mathbf{c} = \mathbf{A}^T \mathbf{w}_*$,

$$\mathbf{w}_* = E[\mathbf{x} y_{\tau_i}] = E[\mathbf{A}\mathbf{s}\mathbf{w}_*^T \mathbf{A}\mathbf{s}_{\tau_i}] = \mathbf{A}E[\mathbf{s}\mathbf{s}_{\tau_i}^T]\mathbf{A}^T \mathbf{w}_*, \tag{3.2}$$

where $s_{\tau_i} = s(k - \tau_i)$.

Multiplying both sides of equation 3.1 by nonsingular mixing matrix, we have $\mathbf{A}$,

$$\mathbf{A}^T \mathbf{w}_* = \mathbf{A}^T \mathbf{A} E[\mathbf{s}\mathbf{s}_{\tau_i}^T]\mathbf{A}^T \mathbf{w}_*, \;\; \mathbf{c} = \mathbf{A}\mathbf{A}^T E[\mathbf{s}\mathbf{s}_{\tau_i}^T]\mathbf{c}. \tag{3.3}$$

Hence, taking into account, that $E[\mathbf{x}\mathbf{x}^T] = \mathbf{I}$, we have

$$E[\mathbf{x}\mathbf{x}^T] = \mathbf{A}E[\mathbf{s}\mathbf{s}^T]\mathbf{A}^T = \mathbf{A}\Psi_{i,0}\mathbf{A}^T. \tag{3.4}$$

Because of the decorrelation assumption, $\Psi_{i,0} = E[\mathbf{s}\mathbf{s}^T]$ is diagonal; thus, $\mathbf{A}^T\mathbf{A} = \Psi_{i,0}$ is also diagonal. Since $\Psi_{i,\tau}$ has all its elements null but ψ_{ii,τ_i}, the only solution for equation 3.3, besides the trivial $\mathbf{c} = \mathbf{0}$ (which is not allowed by normalization) is $\mathbf{c} = \beta \mathbf{e}_i$.

4 Results

We have carried out extensive tests to confirm the validity of the developed algorithm. We illustrate the the performance by two examples.

4.1 Example 1. In this example we have selected four statistically independent zero-mean source signals shown in Figure 1. We have mixed them by randomly generated mixing matrices and extracted them one by one using the following time delays (in number of samples)—$\tau_1 = 113$, $\tau_2 = 15$, $\tau_3 = 6$, and $\tau_4 = 11$—where the time delays have been estimated on the basis of autocorrelation functions. We found that the algorithm extracted all source signals successfully.[2]

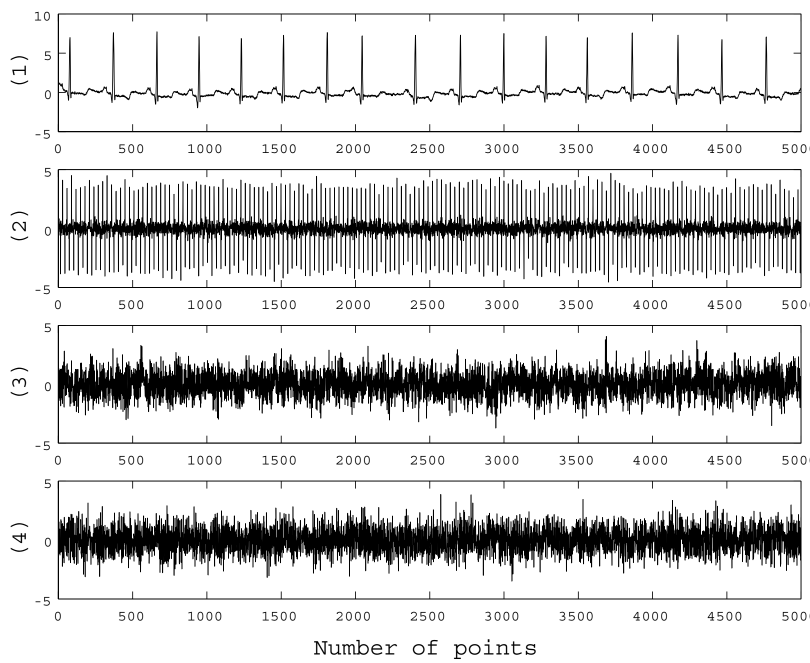

Figure 1: The four signals used in the simulations: (1) Electrocardiographic signal. (2) Spiky signal. (3–4) Two colored gaussian signals with different autocorrelation. Notice that gaussian (zero kurtosis) signals can be separated by this method.

[2] These signals and the algorithm written in MATLAB are available on request.

In fact, in order to check this efficiency, we ran the algorithm 100 times by mixing the sources by a random matrix each time. The time lag τ_i ($i = 1, 2, 3, 4$) was assigned randomly to one of the four at each trial. For example, for τ_2, the spiky signal should be extracted, and so on.

We calculated the number of iterations until the weights achieved convergence and the error between the estimated and the original ith source signal assigned by τ_i. Because these two signals had variance normalized to unity, we calculated the error at each trial q as

$$error(q) = std\{\hat{s} - sign[corr(\hat{s}, s_i)]s_i\}, \tag{4.1}$$

where *std*, *sign*, and *corr* stand for standard deviation, sign, and correlation between two signals, respectively.

The result of the 100 trials were as follows:

$$\text{Overall error} = 0.05 \pm 0.02,$$

$$\text{Number of iterations} = 6.09 \pm 2.19.$$

4.2 Example 2. In this example we tested our algorithm for real-world data—the well-known electrocardiogram (ECG) measured from a pregnant woman and distributed by De Moor (1997).[3] A plot with the input data is shown in Figure 2, where one can see the heart beating of both the mother (stronger and slow) and the fetus (weaker and faster). The task was to extract the cardiac variations of the fetus (FECG).

It was important to calculate the appropriate delay p. We suggested that it could be carried out by examining the autocorrelation $\varsigma(p)$. In Figure 3, this function is shown for the case of the first measured signal, where the fetal influence was clearly stronger than in the other signals. One can see in Figure 3 that $\varsigma(p)$ presents many peaks. Since we do not know which is the most appropriate, it is important to have a previous idea of the most probable value of p. In this case, one could make use of either the fact that a fetal heart rate is around 120 beats per second or by visual inspection examine the timing of the R-R interval. A heart rate of 120 beats per second means that the heart should strike every 0.5 second. By examining the autocorrelation, we found that it had a peak at 0.448 second, corresponding to $p = 112$ samples. Using this information as the time delay for the algorithm, we obtained the FECG signal as shown in the bottom of Figure 2.

[3] Although in his homepage, De Moor assures that the sampling frequency was 500 Hz, we believe that it is most likely to be 250 Hz, which is more coherent with the data; otherwise, the mother's heart beat would be too fast. Therefore, we use this sampling frequency here.

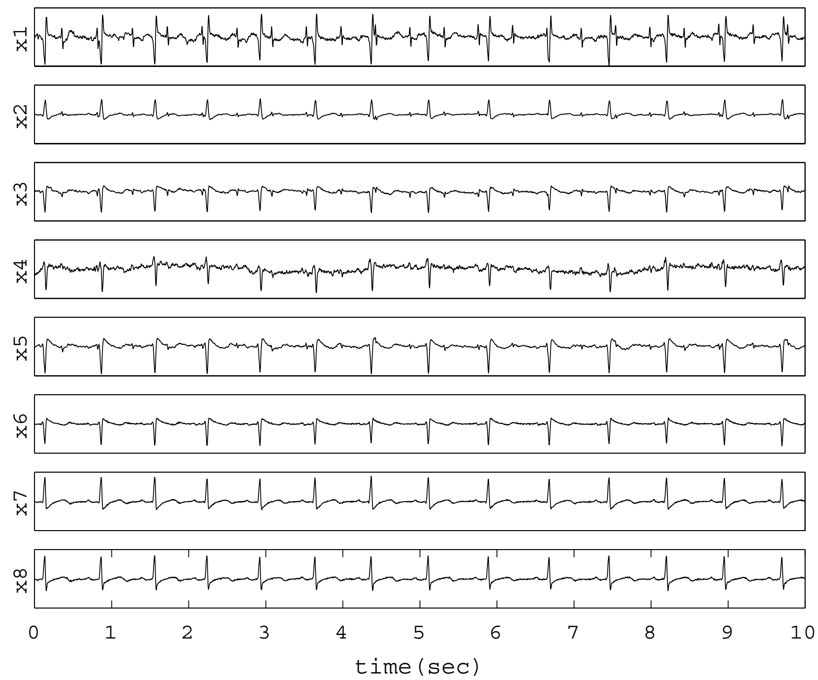

Figure 2: ECG signals taken by abdominal (x1–x5) and thoracic(x6–x8) measurements from a pregnant woman. Notice that the fetal influence is stronger in the first signal.

5 Discussion

The simulation results show that the algorithm worked efficiently, as evidenced by the small overall mean error and its corresponding standard deviation. As predicted by the theory, the algorithm in all the cases extracted the desired source, no matter if it was colored gaussian or not, independent of the mixing matrix. Indeed, it is important to emphasize that this algorithm extracts signals as long as they are decorrelated and show a temporal structure. It is also worth noting that the speed of convergence was fairly high (only a few iterations were needed to achieve convergence), as shown in the experimental results. Moreover, we have shown an example where the algorithm estimated efficiently the fetal ECG from measurements taken from a pregnant woman. We believe it can be useful in other applications too. For example, in biomedical signal processing, especially in EEG/MEG, some oscillatory artifacts such as power supply interference (50–60 Hz), and cardiac artifacts, the optimum time delay can be easily estimated. Similarly,

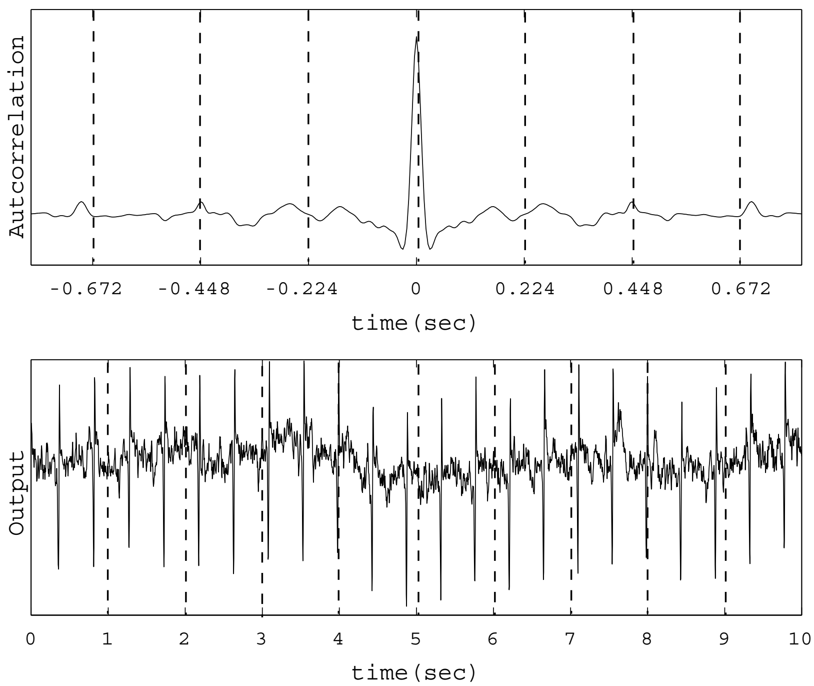

Figure 3: (Top) Autocorrelation of signal x1 of Figure 2. Notice the peak at 0.448 second. (Bottom) Output of the algorithm using the above delay. Notice that it is clearly at a higher rate, compared to the mother, as the fetal ECG should be.

for some experiments with event-related potentials, the time delay can be also found. Moreover, for speech signals linearly mixed with environmental acoustic noise, time delay can be estimated in some parts of spoken speech or music. But this particular problem requires deeper studies, involving the convolution effect.

Even if the remaining source signals do not present the required autocorrelation property, we can extract the ones that present and remove them from the measured signals by using a deflation technique. Examples of how it can be carried out can be found in Delfosse and Loubaton (1995) and Hyvarinen and Oja (1997).

Acknowledgments

We are indebted to Mark Girolami, Paisley University, Scotland for his helpful comments and suggestions.

References

Amari, S. (1999). ICA of temporally correlated signals—learning algorithm. In *Proc. ICA '99* (pp. 13–18). Aussois, France.

Amari, S., & Cichocki, A. (1998). Adaptive blind signal processing—neural network approaches. *Proceedings IEEE* (invited paper), 86(10), 2026–2048.

Barros, A. K., & Ohnishi, N. (1999). Removal of quasi-periodic sources from physiological measurements. In *Proc. ICA '99* (pp. 185–189). Aussois, France.

Barros, A. K., Vigário, R., Jousmäki, V., & Ohnishi, N. (2000). Extraction of event-related signals from multi-channel bioelectrical measurements. *IEEE Trans. on Biomedical Engineering, 47*(5).

Belouchrani, A., Meraim, K., Cardoso, J.-F., & Moulines, E. (1997). A blind source separation technique based on second order statistics. *IEEE Trans. on Signal Processing, 45*, 434–444.

Cichocki, A., Barros, K. (1999). Robust batch algorithm for sequential blind extraction of noisy biomedical signals. In *Proc. ISSPA'99*. Australia.

Cichocki, A., Thawonmas, R., & Amari, S. (1997). Sequential blind signal extraction in order specified by stochastic properties. *Electronics Letters, 33*(1), 64–65.

Comon, P. (1994). Independent component analysis, a new concept? *Signal Processing, 24*, 287–314.

Delfosse, N., & Loubaton, P. (1995). Adaptive blind separation of independent sources: A deflation approach. *Signal Processing, 45*, 59–83.

De Moor, D. (Ed.) (1997). Daisy: Database for the identification of systems. Available online at: http://www.esat.kuleuven.ac.be/sista/daisy.

Hyvarinen, A., & Oja, E. (1997). A fast fixed-point algorithm for independent component analysis. *Neural Computation, 9*, 1483–1492.

Ikeda, S., & Murata, N. (1999). A method of ICA in time frequency domain. In *Proc. ICA '99* (pp. 365–370). Aussois, France.

Jutten, C., & Hérault, J. (1988). Independent component analysis versus PCA. *Proc. EUSIPCO* (pp. 643-646).

Lee, T-W. (1998). *Independent component analysis*. Norwell, MA: Kluwer.

Luo, J., Hu, B., Ling, X-T., & Liu, R-W. (1999). Principal independent component analysis. *IEEE Trans. on Neural Networks, 10*(4), 912–917.

Papoulis, A. (1991). *Probability, random variables, and stochastic processes*. New York: McGraw-Hill.

Received December 29, 1999; accepted November 3, 2000.