SECOND ORDER HEBBIAN NEURAL NETWORKS AND BLIND SOURCE SEPARATION

Konstantinos I. Diamantaras
Department of Informatics
Technological Education Institute, 540 01 Thessaloniki, Greece
Tel: +30 31 791287; fax: +30 31 791290
e-mail: kdiamant@alpha.it.teithe.gr

ABSTRACT

The adaptive blind source separation problem has been traditionally dealt with the use of nonlinear neural models implementing higher-order statistical methods. In this paper we show that second order Cross-Coupled Hebbian rule used for Asymmetric Principal Component Analysis (APCA) is capable of blindly and adaptively separating uncorrelated sources. Our method enjoys the following advantages over similar higher-order models such as those performing Independent Component Analysis (ICA): (a) the strong independence assumption about the source signals is reduced to the weaker uncorrelation assumption, (b) there is no constraint on the sources pdf's, i.e. we remove the assumption that at most one signal is Gaussian, and (c) the higher order statistical optimization methods are replaced with second order methods with no local minima, and(d) the kurtosis of the sources becomes irrelevant. Simulation experiments shows that the model successfully separates source images with kurtoses of different signs.

1 Introduction

During the past decade, the relationship between neural networks implementing linear input-output mappings and standard second-order statistical methods, such as Principal Component Analysis (PCA), has been known and studied extensively [7]. More recently, there is an increased interest on nonlinear neural models which are related to higher-order statistical techniques and are applied in blind inverse problems such as *Blind deconvolution* [9], *Blind source separation* [11], and *Independent Component Analysis (ICA)* [5]. These problems have important applications in the areas of digital communications, and signal processing (e.g. in the processing of biological signals such as ECG, EEG and MEG).

The common framework behind all these problems involves the recovery of hidden stochastic signals (called *sources*) given their observed transformation with some unknown linear operator. For the sake of clarity let us focus on the Blind Source Separation (BSS) problem, although a similar mathematical development holds for the blind deconvolution problem as well. If one uses methods based on higher-order statistics (HOS) one assumes that the source samples are independent random variables and that their pdf's are not Gaussian except for perhaps only one source signal. The problem solution using this approach is based on the extremization of certain indices called *contrast functions* [5] under certain second order constraints. Typically, the contrast function involves the fourth-order cumulant (the *kurtosis*) of the reconstructed signals whereas the second order constraint involves the covariance of these signals [5].

Recently, it was found that the BSS problem can be attacked using second-order methods as well (see for example, [2]). The advantages of this approach are manyfold: (a) the source samples may come simply from uncorrelated variables instead of independent ones, (b) there is no constraint on the signal Gaussianity, and (c) the optimization method used is the well-known eigenvalue decomposition and not some nonlinear optimization technique plagued by local minima and / or slow convergence.

Many neural BSS methods have been proposed in the past. The model of Jutten & Herault [11] involves linear units with feed-back connections. Partly, due to its heuristic origin the model has poor performance and it fails to separate more than two sources. Other models include the Nonlinear and Robust PCA models of Oja and co-workers (for a good review see [12]), the Equivariant Adaptive algorithm [3], the entropy maximization model [1], the fixed-point algorithm [10], the robust adaptive algorithm [4], the exploratory projection pursuit (EPP) network [8] etc. The common characteristic of these methods is the fact that they adaptively extremize some higher-order statistical index, and as a result they inherit the assumptions and drawbacks of HOS-based methods. For example, these models assume that the data samples come from independent random variables, that at most one source is Gaussian, and most of the models assume that the source signal kurtoses are of the same sign (usually negative).

The use of second-order time-delayed neural models for blind source separation was proposed by Molgedey and Schuster [13]. Their model incorporates linear units and it concurrently extracts all independent sources assuming prior knowledge of their number. However, the model is recurrent so that the input-output mapping is given through a matrix inversion operation, and the learning algorithm is given in the form of a differential equation which can not be easily transformed into the form of a discrete-time updating equation.

In this paper we show how the cross-coupled Hebbian learning rule originally proposed for Asymmetric Principal Component Analysis (APCA) can be used for solving the blind source separation problem. The rule can extract one source at a time until there is no more variance left in the signal. In particular, Section 2 reviews the use of second order statistics for the blind separation of uncorrelated sources based on similar work proposed for digital communications problems. After using a simple neural orthogonalization procedure in Section 3, Section 4 describes the Asymmetric PCA problem and the neural algorithm used for solving it. In Section 5 we combine the above neural models into an adaptive algorithm for blind source separation. Finally Section 6 contains results from simulation experiments which used our neural method for the efficient separation of image mixtures with sources of different kurtosis signs.

2 Second-order blind separation of sources

Consider the mixing of n unknown continuous-time signals $s_i(k)$, $i = 1, \dots, n$, through an equally unknown linear mixing operator $\mathbf{A}$

$$\mathbf{x}(k) = [x_1(k) \cdots x_n(k)]^T = \mathbf{A}\mathbf{s}(k). \tag{1}$$

The number of observation signals is equal to the number of sources. Our assumptions regarding the source signals and the mixing matrix are summarized below:

A.1 The sources are at least wide-sense stationary and they are pairwise statistically orthogonal, i.e. if $i \neq j$ then $E\{s_i(k_1)s_j(k_2)\} = 0$, for all k_1, k_2.

A.2 $\mathrm{rank}(\mathbf{A}) = n$.

The recovery of the sources $s_i(k)$ when only the observation process $\mathbf{x}(k)$ is available is called the *Blind Source Separation (BSS)* problem. The BSS problem has inherent indeterminacies due to the lack of knowledge about the structure of $\mathbf{A}$: it is impossible to recover the original amplitude and ordering of the sources.

In this paper we shall use second order statistics assuming that the source signals have colored spectra. In particular, let us define the source autocorrelation matrix for some time-delay l as

$$\mathbf{R}_s(l) = E\{\mathbf{s}(k)\mathbf{s}(k+l)^T\}.$$

The basic assumptions *A.1* and *A.2* are complemented with the following

A.3 The matrices $\mathbf{R}_s(0)$ and $\mathbf{R}_s(l)$, for some $l \geq 1$, are non-zero. Due to the pairwise orthogonality of the sources, it follows that $\mathbf{R}_s(0)$ and $\mathbf{R}_s(l)$ are *diagonal* matrices, whose diagonal entries can be written as: $[\mathbf{R}_s(0)]_{ii} = Es_i^2(k) \neq 0$, $[\mathbf{R}_s(l)]_{ii} = Es_i(k)s_i(k+l) \neq 0$. Furthermore, we shall assume that $\mathbf{R}_s(l)$ has non-repeated diagonal elements.

A.4 Due to the inherent amplitude indeterminacy of the blind source separation problem we may assume, without loss of generality, that the source signals are normalized to unit variance $\mathbf{R}_s(0) = \mathbf{I}$.

Using the assumptions A.1 to A.4 a second-order BSS algorithm is described by the following four steps (based on the work of Belouchrani e.a. [2]):

Step 1: Eigenvalue decomposition.

$$\mathbf{R}_x(0) = \mathbf{U}_0\mathbf{\Lambda}_0\mathbf{U}_0^T \tag{2}$$

Step 2: Orthogonalization. The observation data $\mathbf{x}(k)$ are orthogonalized using a linear transform $\mathbf{z}(k) = \mathbf{C}\mathbf{x}(k)$, where $\mathbf{C} \equiv \mathbf{U}_0\mathbf{\Lambda}_0^{-1/2}\mathbf{U}_0^T = \mathbf{R}_x(0)^{-1/2}$. This transformation creates a new effective mixing matrix $\tilde{\mathbf{A}} = \mathbf{C}\mathbf{A}$, whose rows are orthonormal vectors since $\mathbf{R}_z(0) = E\{\mathbf{z}(k)\mathbf{z}(k)^T\} = \tilde{\mathbf{A}}\mathbf{R}_s(0)\tilde{\mathbf{A}}^T = \tilde{\mathbf{A}}\tilde{\mathbf{A}}^T = \mathbf{I}$.

Step 3: Eigenvalue Decomposition. Choose some $l \geq 1$ such that $\mathbf{R}_s(l) \neq 0$. We have

$$\mathbf{R}_z(l) = \tilde{\mathbf{A}}\mathbf{R}_s(l)\tilde{\mathbf{A}}^T, \tag{3}$$

where $\mathbf{R}_s(l)$ is diagonal, and $\tilde{\mathbf{A}}$ is orthogonal. It follows that (a) Eq. (3) is an eigenvalue decomposition of the matrix $\mathbf{R}_z(l)$ and (b) the matrix is symmetric. By the assumption that the diagonal entries of $\mathbf{R}_s(l)$ are distinct, the eigenvalue decomposition is unique up to a permutation of the eigenvectors. Therefore, if

$$\mathbf{R}_z(l) = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \tag{4}$$

is an eigenvalue decomposition of $\mathbf{R}_z(l)$ then we have a perfect estimate of $\tilde{\mathbf{A}}$ by $\hat{\mathbf{A}} = \mathbf{U}$ (up to a permutation of its columns which corresponds to a permutation of the source signal ordering).

Step 4: Source Estimation. Since $\hat{\mathbf{A}}$ is orthogonal, the source estimates are $\hat{\mathbf{s}}(k) = \hat{\mathbf{A}}^T\mathbf{z}(k)$.

Notice that if we have perfect estimates of $\mathbf{R}_x(0)$ and $\mathbf{R}_x(l)$ the reconstruction is also perfect –within the inherent problem limitations.

3 Adaptive orthogonalization.

The first step of our adaptive second order BSS algorithm is the orthogonalization of the observed mixtures. To that end we may use a simple linear feed-forward network as proposed by Girolami and Fyfe [8]. If we let $z_i = \sum_j c_{ij} x_j$ be the network outputs, then the orthogonalizing learning rule is

$$\mathbf{C}_{k+1} = \mathbf{C}_k + \beta_k(\mathbf{I} - \mathbf{z}(k)\mathbf{z}(k)^T) \tag{5}$$

The step-size parameter β_k is set to decrease gradually to zero. The rule (5) can be shown to converge to the symmetric, inverse square root of the data covariance matrix: $\mathbf{C}_\infty = \mathbf{R}_x(0)^{-1/2}$.

4 Asymmetric principal component analysis (APCA).

Asymmetric Principal Component Analysis (APCA) [7] is an extension of PCA involving two stochastic processes. Ordinary PCA can be viewed in two equivalent ways: (a) as a method which maximizes the signal projection variance, or (b) as a method which minimizes the approximation error. Therefore, there are two independent ways in which to extend PCA when we have two vector stochastic signals, $\mathbf{x}$ and $\mathbf{y}$: one way is along the covariance direction and one along the approximation direction.

The relevant extension here is the one according to the variance criterion which leads to the *Cross-correlation APCA problem* [7]. The problem amounts to maximizing the cross-correlation index between the two random vectors $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$,

$$J(\mathbf{U},\mathbf{V}) = E\Big\{\mathrm{tr}[(\mathbf{U}\mathbf{x})(\mathbf{V}\mathbf{y})^T]\Big\} = \mathrm{tr}(\mathbf{U}\mathbf{R}_{xy}\mathbf{V}^T), \tag{6}$$

under the constraint $\mathbf{U}\mathbf{U}^T = \mathbf{V}\mathbf{V}^T = \mathbf{I}$, where $\mathbf{U} \in \mathbb{R}^{p\times n}$, $\mathbf{V} \in \mathbb{R}^{p\times m}$, $p < \min\{m,n\}$. The solution to the problem is the SVD of the cross-correlation matrix $\mathbf{R}_{xy} \equiv E\{\mathbf{x}\mathbf{y}^T\}$.

The neural implementation of cross-correlation APCA has been proposed in conjunction with the *cross-coupled Hebbian rule* [6]. Let $a = \mathbf{u}^T\mathbf{x}$, $b = \mathbf{v}^T\mathbf{y}$, be the output activations of two *linear* neurons. The cross-coupled Hebbian rule is described by the recursive equations

$$\Delta\mathbf{u}_k = \beta_k[\mathbf{x}(k) - a(k)\mathbf{u}_k]b(k) \tag{7}$$
$$\Delta\mathbf{v}_k = \beta_k[\mathbf{y}(k) - b(k)\mathbf{v}_k]a(k) \tag{8}$$

The rule has been shown to converge to the principal singular vectors of $\mathbf{R}_{xy}$. The second, third, etc, components can be extracted using the *Deflation Transformation*:

$$\mathbf{x}(k)' \leftarrow [\mathbf{I} - \mathbf{u}\mathbf{u}^T]\mathbf{x}(k) \tag{9}$$
$$\mathbf{y}(k)' \leftarrow [\mathbf{I} - \mathbf{v}\mathbf{v}^T]\mathbf{y}(k) \tag{10}$$

where $\mathbf{u}$, $\mathbf{v}$ are the latest extracted unit-length singular vectors. It can be shown that the singular value decomposition of $\mathbf{R}_{x'y'}$ for the new signals $\mathbf{x}(k)'$, $\mathbf{y}(k)'$, is the same as the SVD of the original matrix $\mathbf{R}_{xy}$, except for the singular value corresponding to $\mathbf{u}$, $\mathbf{v}$, which has become equal to zero. Therefore, after extracting the principal singular vectors, if we perform the deflation transformation on the data and we run the algorithm (7), (8), again we shall obtain the second principal singular vectors. In that way we can recursively extract all the singular vectors of the original covariance matrix $\mathbf{R}_{xy}$.

5 Neural, second-order blind source separation

Putting all the pieces together we describe in this section the complete adaptive second-order BSS algorithm. In the beginning the data are orthogonalized using the adaptive rule (5) obtaining the vector sequence $\mathbf{z}(k)$. The neural cross-correlation APCA model in Section 4 can be applied directly for computing the SVD of the matrix $\mathbf{R}_z(l) = E\mathbf{z}(k)\mathbf{z}(k+l)^T$ by setting $\mathbf{x}(k) = \mathbf{z}(k)$ and $\mathbf{y}(k) = \mathbf{z}(k+l)$ in Equations (7) and (8). So,

$$\Delta\mathbf{u}_k = \beta_k[\mathbf{z}(k) - a(k)\mathbf{u}_k]b(k) \tag{11}$$
$$\Delta\mathbf{v}_k = \beta_k[\mathbf{z}(k+l) - b(k)\mathbf{v}_k]a(k) \tag{12}$$
$$a(k) = \mathbf{u}_k^T\mathbf{z}(k) \tag{13}$$
$$b(k) = \mathbf{v}_k^T\mathbf{z}(k+l) \tag{14}$$

Note that $\mathbf{R}_z(l)$ is (ideally) symmetric so the left and right singular vectors are identical up to their sign. Therefore, for the convergence points of Algorithm (11), (12) we must have $\mathbf{u}_\infty = \pm\mathbf{v}_\infty$. In practice, the previous equality does not hold perfectly and we may prefer to choose either vector u_∞ or $\mathbf{v}_\infty$ as the final solution (or the average of the two).

After computing each pair of singular vectors, we deflate the data using one of the following two transforms $\mathbf{x}'(k) = [\mathbf{I} - \mathbf{u}\mathbf{u}^T]\mathbf{x}(k)$ or $\mathbf{x}'(k) = [\mathbf{I} - \mathbf{v}\mathbf{v}^T]\mathbf{x}(k)$. Then we run the algorithm (11), (12) again on the new data obtaining recursively the third, fourth, etc, singular vectors.

6 Simulation experiments

We tested the method using four source images treated as signals by stacking the columns on top of each other into a long data vector. Both the original images and their linear mixtures are shown in Figure 1, whereas the unmixing performed by the neural algorithm is shown in Figure 2. We note that image sources s1, s2, and s3 have negative kurtoses, while image s4 has positive kurtosis. As a result, most higher order methods would not work in this case since they assume that all sources have kurtoses of the same sign.

7 Conclusion

In this paper we presented a neural second-order approach for blindly unmixing uncorrelated signals. Our method is a combination of an adaptive orthogonalization model and the cross-coupled Hebbian model for Asymmetric PCA. The model does not use any ad-hoc nonlinear unit activation functions, the sources may be simply uncorrelated and not necessarily independent, while we remove all constraints on the signal pdf's. Finally, the kurtosis of the mixed signals is irrelevant here as proven by simulation experiments which successfully separate source images with kurtoses of different signs.

References

[1] A. J. Bell and T. J. Sejnowski, "An Information-Maximization Approach to Blind Separation and Blind Deconvolution". *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995

[2] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A Blind Source Separation Technique Using Second-Order Statistics", *IEEE Tr. Signal Processing*, vol. 45, no. 2, pp. 434–444, Feb. 1997.

[3] J.-F. Cardoso and B. H. Laheld, "Equivariant Adaptive Source Separation". *IEEE Tr. Signal Processing*, vol. 44, no. 12, pp. 3017–3030, Dec. 1996.

[4] A. Cichocki, S. Amari, M. Adachi and W. Kasprzak, "Self Adaptive Neural Networks for Blind Separation of Sources", in *Proc. Int. Symp. Circuits & Systems, ISCAS-96*, pp. 157–161, Atlanta, GA, 1996.

[5] P. Comon, "Independent Component Analysis, A New Concept?", *Signal Processing*, vol. 36, no. 3, pp. 287–314, 1994.

[6] K. I. Diamantaras and S. Y. Kung, "Cross-correlation Neural Network Models", *IEEE Tr. Neural Networks*, vol. 42, no. 11, Nov. 1994.

[7] K. I. Diamantaras and S. Y. Kung, *Principal Component Neural Networks: Theory and Applications*, Wiley Interscience, N.Y., 1996.

[8] M. Girolami and C. Fyfe. "An Extended Exploratory Projection Pursuit Network with Linear and Nonlinear Anti-Hebbian Lateral Connections Applied to the Cocktail Party Problem". *Neural Networks*, 10(9):1607–1618, 1997.

[9] S. Haykin (ed.), *Blind Deconvolution.* Prentice Hall, NJ, 1994.

[10] A. Hyvärinen and E. Oja, "A Fast Fixed-Point Algorithm for Independent Component Analysis", *Neural Computation*, vol. 9, no. 7, pp. 1483-1492, 1997.

[11] C. Jutten and J. Herault, "Blind Separation of Sources, Part I: An Adaptive Algorithm Based on Neuromimetic Architecture", *Signal Processing*, vol. 24, no. 1, pp. 1–10, July 1991.

[12] J. Karhunen, E. Oja, L. Wang, R. Vigario, and J. Joutsensalo, "A Class of Neural Networks for Independent Component Analysis", *IEEE Tr. Neural Networks*, vol. 8, no. 3, pp. 486–504, May 1997.

[13] L. Molgedey and H. G. Schuster, "Separation of a Mixture of Independent Signals Using Time Delayed Correlations", *Physical Review Letters*, vol. 72, no. 23, June 1994.

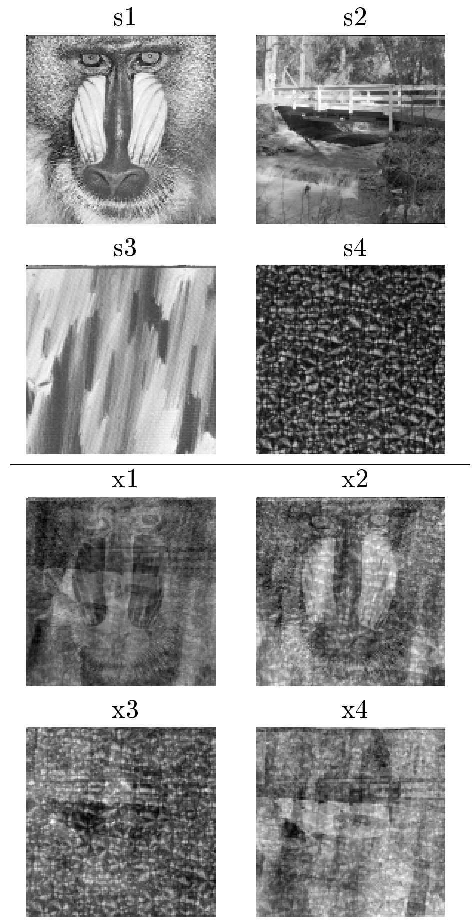

Figure 1: The source images (top 2 rows) and their linear mixtures (bottom 2 rows). Sources s1, s2, s3 have negative kurtoses, while s4 has positive kurtosis. As a result, most higher order neural models are not applicable in this case.

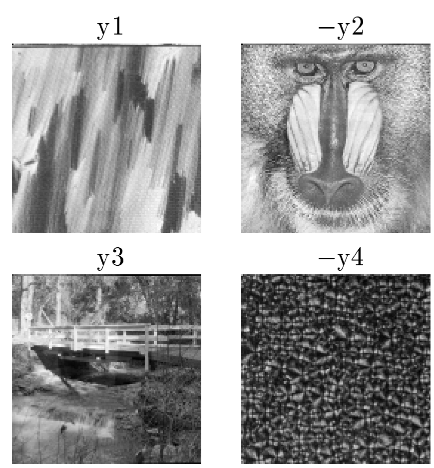

Figure 2: The unmixing results obtained by the neural algorithm.