Adaptive-Geometric methods: application to the separation of EEG signals

C.G.Puntonet *, C.Bauer *, **, E.W.Lang **, M.R.Alvarez *, B.Prieto *

* Dept. of Architecture and Computer Technology. University of Granada (Spain)
** Institute of Biophysics. University of Regensburg (Germany)

ABSTRACT

This paper presents a new adaptive algorithm for the on-line linear and non-linear separation of signals with non-uniform, symmetrical probability distributions. The procedure is based on the interpretation and properties of the vectorial spaces of sources and mixtures, using a multiple linearization in the mixture space. The main characteristics of the procedure are its simplicity, its immunity to symmetrically-distributed additive noise, and the rapid convergence experimentally validated when the method is applied to the separation of multiple EEG signals.

1. INTRODUCTION

The problem of linear blind separation of sources involves obtaining the signals generated by *p* sources, vectorially represented by $\mathbf{s}(t)=[s_1(t),....,s_p(t)]^T$, from the linear mixture signals, $\mathbf{e}(t)=[e_1(t),...e_p(t)]^T$. The mixture, normally produced in a medium or in the sensors, is characterized by a non singular matrix **A** such that $\mathbf{e} = \mathbf{A}\ \mathbf{s}$. The goal traditionally sought within the context of separation of sources is to estimate **A** by means of another matrix **W** such that the output vector $\mathbf{s}^* = \mathbf{W}^{-1}\mathbf{e}$ coincides with that of the original sources except for a scale factor and a permutation, i.e.

$$W^{-1} = A\ P\ I \tag{1}$$

where **P** is a permutation matrix and **D** is a diagonal matrix. Any matrix **W** related to **A** as in (1) is said to be *similar to* **A**.

For linear mixtures and bounded sources, we have presented a new way to apply the geometrical method, which does not require obtaining vertices or edge points. From geometric considerations, it is possible to obtain a matrix that is similar to **A** by determining the slopes of, or any vector on, the edges that are incident on any one of the vertices. We have presented various algorithms [4, 6], all of which attempt to select one vertex, from the mixing vectors obtained, and one or more points from each of the edges that are incident to it. The problem is then reduced to that of finding the distribution axes of probability within the known mixing space, as these are parallel to the axes of the hyperparallelepiped that contains the observation space [5].

An interesting way was recently proposed to solve the separation of linear mixtures, consisting of using the Kullback-Leibler divergence or the mutual information [3], where the independent axes are computed efficiently. For post-nonlinear mixtures, a batch procedure based on a maximum likelihood approach has been developed in [8]. For non-linear mixtures, the method proposed in [2] is adaptive and adequate for sharply peaked distributions. The approach presented here is related to the above, but is valid for more than two signals and for “non-symmetric” non-linear mixtures.

In [7] we described an adaptive procedure for the demixing of linear and non-linear mixtures of two signals with probability distributions that are symmetric with respect to their centres and non uniform, performing a fixed piecewise linearization in the case of nonlinear mixtures in order to obtain the distribution axes of probability that are parallel to the slopes of the parallelepiped for two sources.

Here, we combine the geometric properties of the distributions, which provide the independent components, with the advantages of the competitive neural networks, in order to adaptively compute the axes of the density function in the linear or non-linear observation space, by means of a dynamic piecewise linearization valid for all kinds of sources exhibiting an unimodal probability distribution, such as Gaussian, Laplacian, Poisson, Gamma and others.

2. PROPOSED METHOD

The main idea of this method combine geometric considerations, previously developed, with an adaptive processing, and it consists of normalizing the observed space in a set of concentric layers in order to adaptively compute the slopes corresponding to the independent axes of the distributions of the mixtures by means of an array of symmetrically distributed neurons in each dimension. The procedure has two stages, firstly a preprocessing stage to normalize the observed space, followed by the processing or learning of the neurons, which estimate the high density regions in a similar, but not identical way to that of self organizing maps.

2.1 Preprocessing

First of all, the observed signals e´ are set to zero mean, μ, and unity variance, σ, as follows:

$$\boldsymbol{e} = \frac{(\boldsymbol{e}' - \mu)}{\sigma} \tag{2}$$

The average correlation coefficient is also computed by means of the following expression:

$$<\varrho> = \frac{1}{p}\sum_{i,j}\varrho_{ij} \quad , \quad \varrho_{ij} = \frac{1}{T}\sum_{t=1}^{T}(e_i(t).e_j(t)) \qquad i,j \in \{1...p\} \quad , \quad i<j \tag{3}$$

From (3), the procedure can eliminate unnecessary samples by using the parameter δ, as follows:

$$\delta = \exp(-<\varrho>^2) \tag{4}$$

Many kind of signals, as speech or EEG signals, present unnecessary points near the origin that do not provide information when the computation of the distribution axes is being carried out (Figure 1), and these can be adecuately removed if the following condition is verified:

$$\|\boldsymbol{e}\| < \sum_i \sigma_i . \delta = R \quad , \quad i \in \{1...p\} \tag{5}$$

where R is the radius of the eliminated points circle. Moreover, the algorithm can select useful points near the independent components (Figure 2), as follows:

$$e_k < \sigma_k \, \delta \quad , \quad \forall k \in \{1,...,p\} \tag{6}$$

2.2 Processing

The observation space is subsequently quantized in n circles or layers each with a radius ρ(k) (k=1...n) covering the points as follows:

$$\exists \, \rho(k) \mid \rho(k-1) < \|\boldsymbol{e}(t)\| < \rho(k) \quad , \quad \forall k \in \{1,...,n\} \tag{7}$$

The distance between one point and the 2p neurons existing in the p-dimensional space is:

$$d(i,\rho(k)) = \| \boldsymbol{w}_i(\rho(k),t) - \boldsymbol{e}_i(\rho(k),t) \| \qquad i \in \{1,...,2p\} \quad , \quad k \in \{1,..,n\} \tag{8}$$

A winner neuron, labeled i^*, in a layer ρ(k), has a minimum distance to the point and verifies:

$$d(i^*,\rho(k)) = \min\{d(i,\rho(k))\} \; i \in \{1,.,2p\} \, k \in \{1,.,n\} \tag{9}$$

From now on, we will denote ρ the layer ρ(k) defined in (7). After this, three methods can be used for the learning process. If a winner neuron approaches the density region, at time t, the learning rule is given by:

$$w_{i^*}(\rho,t+1) = w_{i^*}(\rho,t) + \alpha(t) \; sgn(e_i(\rho,t) - w_{i^*}(\rho,t)) \tag{10}$$

where:

$$\alpha(t+1) = \frac{\alpha(t) \; \rho \; \delta}{1+\beta(i^*,\rho)^{-2}} \qquad i^* \in \{1,...,2p\}$$
$$\beta(i^*,\rho) = <sgn(e_{i^*}(\rho,t) - w_{i^*}(\rho,t)> \tag{11}$$

where α is a geometry-dependent decreasing learning rate. For the sake of simplicity, equation (10) does not show the inverse movement of the symmetric neuron (ii*) belonging to the same axis, and this property decreases the convergence time. Note that a great variety of suitable monotonic functions, α, can be used.

A second type of learning, similar to the first, can also be used in which the learning space of each neuron, i, is reduced to its associate quadrant, c; this is useful when it is known in certain real applications that the mixing matrix, **A**, verifies $a_{ii} > a_{ij}$. If this is so, only the representative winner neuron, ic*, is active, and it is only necessary to detect the quadrant, c, e(ρ,t) belongs to, as follows:

$$w_{ic^*}(\rho,t+1) = w_{ic^*}(\rho,t) + \alpha(t) \; sgn[e_i(\rho,t) - w_{ic^*}(\rho,t)] \qquad ic^* \in \{1,...,2p\} \tag{12}$$

The third learning procedure activates all neurons at once, by means of a factor, k(t), that modulates competitive learning, as in self-organizing systems.

$$w_i(\rho,t+1) = w_i(\rho,t) + \alpha_1(t) \; sgn[e_i(\rho,t) - w_{i^*}(\rho,t)] \; k(t)$$
$$k(t) = \exp(-\alpha_2(t) \; \|e_i(\rho,t) - w_{i^*}(\rho,t)\|^2) \tag{13}$$

Here α_1(t) and α_2(t) are geometry-dependent learning rates. After the learning process, the neurons are maintained on their respective layers by means of the following normalization:

$$w_i(\rho,t) = \frac{w_i(\rho,t) \; \rho}{\|w_i(\rho,t)\|} \quad , \quad i \in \{1..2p\} \;\; \rho \in \{\rho(1)..\rho(n)\} \tag{14}$$

Some improvements have been made to the process for good estimation of the distribution axes. For non-linear mixtures, the spatial neuron order in successive layers may change, and for correct adaptive separation it is necessary to check, periodically, the following expression:

$$\min(i,j) = \min \| w_i(\rho,t) - w_j(\rho-1,t) \| \quad i,j \in \{1...2p\} \qquad (15)$$

Once this expresion is computed, the rearranging is done bottom-up, begining from layer (ρ-1) in (15), if the condition min(i,j) < min(i,i) is verified.

Furthermore, in linear or non-linear mixtures the real observed signals may exhibit non-uniform density distributions, and the procedure generates adaptively variable layers in accordance with the point density. Then, the distance between the circles, ρ(i,τ), in time τ, can be adjusted as a function of the density points, λ(i,τ), between two successive layers:

$$\rho(i,\tau+1) = \rho(i,\tau) + \gamma \, (\lambda(i-1,\tau) - \lambda(i,\tau)) \qquad (16)$$

where γ is a learning rate.

3. SEPARATION MATRICES

Due to the piecewise linearization of the space, a set, **W**, of matrices is obtained for each layer:

$$\boldsymbol{W} = \left\{ W_{\rho(1)} \,,\; \ldots \;,\, W_{\rho(n)} \right\} \qquad (17)$$

where the matrices $W_{\rho(i)}$ are equal to:

$$W_{\rho(i)} = \begin{pmatrix} W_{11\,\rho(i)} & \cdot\cdot & W_{1p\,\rho(i)} \\ W_{p1\,\rho(i)} & \cdot\cdot & W_{pp\,\rho(i)} \end{pmatrix} \qquad (18)$$

For linear systems or "symmetric" non-linear mixtures [2], the elements of the matrix (18) are the symmetric slopes, in the layer ρ(k), between two neurons of each dimension, computed in (10), (12) or (13), i.e.:

$$W_{ij\,\rho(k)} = \frac{w_{2j\;i\;\rho(k)} - w_{2j\;i\;\rho(k-1)}}{w_{2j\;j\;\rho(k)} - w_{2j\;j\;\rho(k-1)}} \qquad (19)$$

$$i,j \in \{1...p\} \;,\; k \in \{1\ldots n\}$$

Since the main simulation presented in this paper refers to a linear mixture of EEG signals, we will use this particular expression (19), although in the general case and for pure non-linear mixtures (without symmetry at the origin), the above expression must be replaced by a similar one, as follows:

$$W_{ij\,\rho(k)} = \frac{w_{\xi(j)\;i\;\rho(k)} - w_{\xi(j)\;i\;\rho(k-1)}}{w_{\xi(j)\;j\;\rho(k)} - w_{\xi(j)\;j\;\rho(k-1)}}$$

$$\xi(j) \in \{ (\alpha,\beta) \mid d(\alpha,\rho) < d(\gamma,\rho) \;,\; d(\beta,\rho) < d(\gamma,\rho) \} \qquad (20)$$

$$i,j \in \{1...p\} \;,\; k \in \{1\ldots n\} \;,\; \alpha\beta\gamma \in \{1\ldots 2p\} \;,\; \alpha \neq \beta$$

Note that (19) is a particular case of (20), with ξ(j)=2j, and that $W_{ii\rho(k)} = 1$ in both equations. Equation (20) means that the subspace associated to the neurons labeled (α,β) around point $e_{i\rho(k)}(t)$ provides the linear contour where the mixture can be considered linear.

For the purpose of separation, the network uses the typical recursive recall taking into account the layer quantization in the observation space, i.e.:

$$s_i(t+1) = e_{i\,\rho(k)}(t) - \sum_{i=1}^{p} w_{ij\,\rho(k)}\, s_j(t) \qquad (21)$$
$$i \neq j \in \{1,...,p\} \;,\; k \in \{1,....,n\}$$

4. SIMULATION RESULTS

Two simulations are presented in order to show the efficiency of the proposed algorithms.
The first one, corresponds to a sinthetic non-linear mixture presented in [2] for sharply peaked distributions, original sources being digital 32-bit signals, as follows:

$$\begin{aligned} e_1(t) &= -2\, sgn[s_1(t)]\; s_1(t)^2 + 1.1\, s_1(t) - s_2(t) \\ e_2(t) &= -2\, sgn[s_2(t)]\; s_2(t)^2 + 1.1\, s_2(t) + s_1(t) \end{aligned} \qquad (22)$$

As shown in Figure 4, good estimation of the density distribution is obtained with 3 iterations of 10000 samples, and using n=4 layers. The four matrices (18) obtained were the following:

$$W_{\rho(1)} = \begin{pmatrix} 1 & 1.7 \\ -1.6 & 1 \end{pmatrix} \quad W_{\rho(2)} = \begin{pmatrix} 1 & 0.25 \\ -0.22 & 1 \end{pmatrix}$$
$$W_{\rho(3)} = \begin{pmatrix} 1 & 0.2 \\ -0.22 & 1 \end{pmatrix} \quad W_{\rho(4)} = \begin{pmatrix} 1 & 0.1 \\ -0.15 & 1 \end{pmatrix} \qquad (23)$$

The second simulation corresponds to a real mixture of 21 EEG signals with 25 time series of 7680 samples each. Similar results as in [1] are obtained. Three artifactual and two tumor related ICA components may be identified from Figure 6. Eye blink artifacts in the EEG data are isolated to ICA component number 2. ICA components numbers 4 and 17 show δ and ϑ waves with a low characteristic frequency and not observed with normal objects. Component number 13 reveals small periodic muscle spiking and any cardiac contamination in the EEG data is concentrated in ICA component number 21. Analyzing the obtained values in (20), the procedure ensures that the hypothese of linear model for this kind of recording is a good approximation.

5. CONCLUSIONS

We have shown the powerful of adaptive-geometric methods by means of a piecewise linearization in the mixture space and using competitive unsupervised learning, in order to find the density function of the observed signals or independent components.
Future work concerns the study of the noise and the implementation of the proposed procedure, once a wide range of non-linear mixtures have been simulated with real signals and using more than two neurons per dimmension.

ACKNOWLEDGEMENTS

This work has been supported in part by the Spanish CICYT proyects TIC98-0982 and TIC97-1149, and a grant of the Cusanuswerk Foundation. We are grateful for the constructive comments and suggestions of Pedro Martín Smith, Alberto Prieto and Matthias Habl.

REFERENCES

[1] M.Habl, Ch.Bauer, Ch.Ziegaus, E.W.Lang, F.Schulmeyer, “Analyzing Brain Tumor related EEG signals with ICA Algorithms”, to appear in Proceedings of the First International Conference on ANN in Medicine and Biology, Goteborg (Sweden), 2000.
[2] J.K. Lin, J.D. Cowan, “Faithful Representation of separable Input Distributions”, Neural Computation, vol. 9, 1997.
[3] D.T. Pham, “Blind Separation of Instantaneous Mixtures of Sources Based on Order Statistics”, IEEE Trans. Signal Processing. (To appear)
[4] A. Prieto, C.G. Puntonet, and B. Prieto, “A neural Learning Algorithm for Blind Separation of Sources Based on geometric Properties”, Signal Processing, vol.64, no.3, pp. 315-331, 1998
[5] A. Prieto, B. Prieto, C.G. Puntonet, A. Cañas, P. Martín-Smith, “Geometric Separation of Linear Mixtures of Sources: Application to Speech Signals”. ICA-99, Aussois (France), Jan 1999.
[6] C.G. Puntonet, and A. Prieto, “Neural Net Approach for Blind Separation of Sources based on Geometric Properties”, Neurocomputing, vol. 18, no.1-3, pp. 141-164, Jan. 1998.
[7] C.G. Puntonet, M.R. Alvarez, A. Prieto, B. Prieto, “Separation of Speech Signals for Nonlinear Mixtures”, Lecture Notes in Computer Science, 1607, Proceedings of IWANN-99, Vol. II, Springer.
[8] A. Taleb, C. Jutten, “Source Separation in Postnonlinear Mixtures. IEEE Trans. S.P., vol. 47, n.10, 1999.

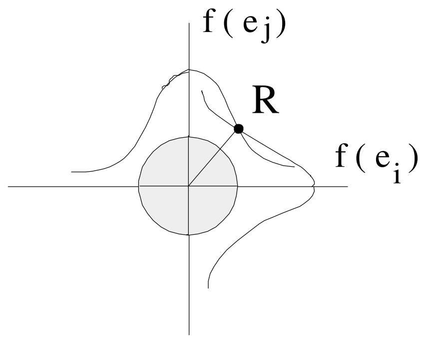

Figure 1. Eliminating points without information

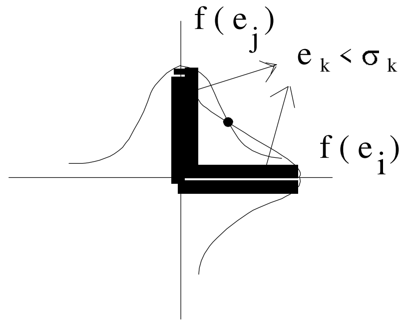

Figure 2. Points near the independent axes

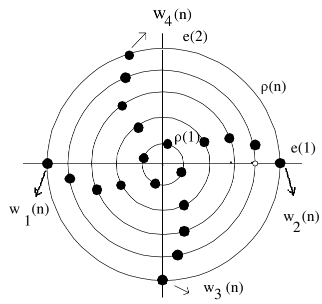

Figure 3. n-layer quantization for p=2.

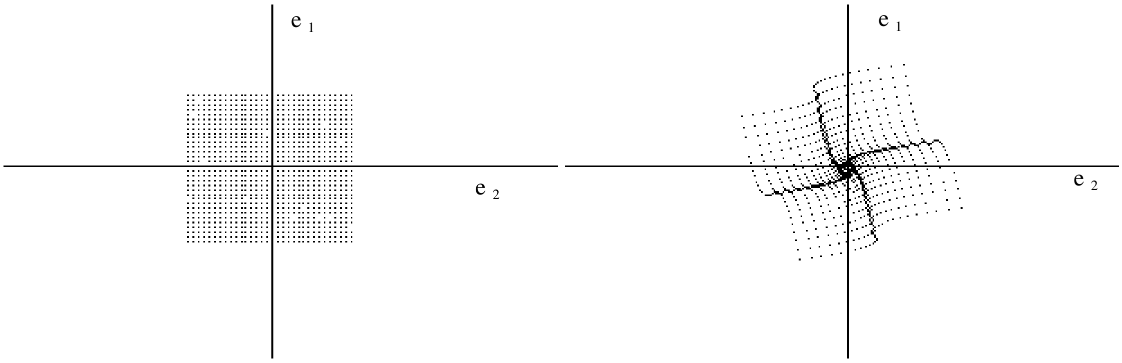

Figure 4. a) Space of original digital 32-valued signals. b) Space of non-linear mixture.

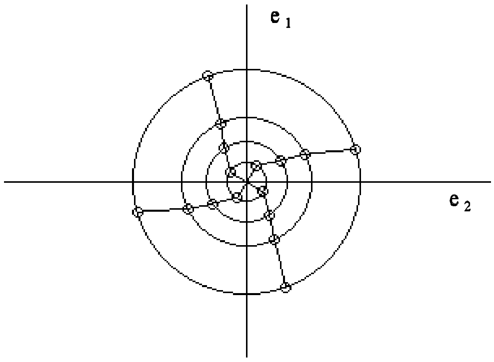

Figure 5. Configuration of the neurons with 3 iterations and 4 layers.

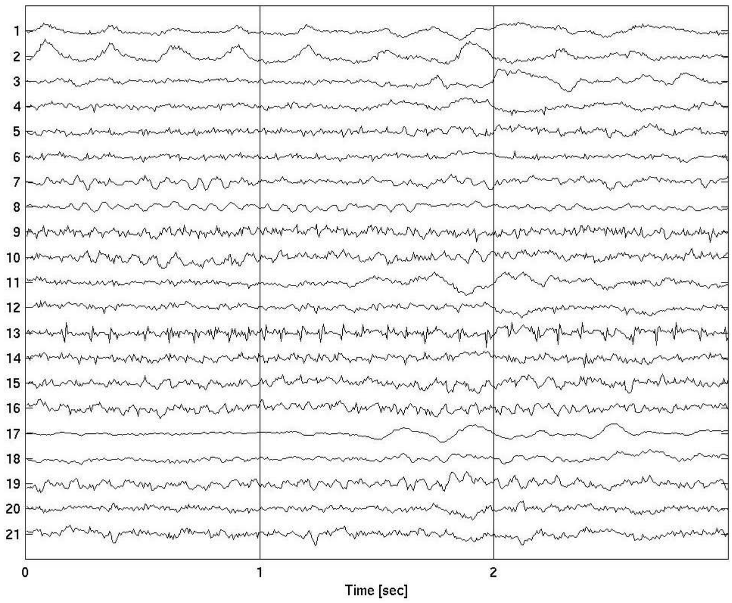

Figure 6. ICA components obtained from EEG sensors