A Stochastic and Competitive Network for the Separation of Sources

C. G. Puntonet*, A. Mansour**, M.R. Alvarez*, B. Prieto*, I. Rojas*

* Dept. of Architecture and Comp. Tech. University of Granada (Spain) ** Bio-Mimetic Control Research Center (RIKEN), Nagoya-shi (Japan)

Spain)
Japan)
Japan)
mear sep
mulated prope
inity the prope
inity of the same of the same
inity of the same of the same of the same
data periods:
data periods: Abstract. This paper presents an adaptive procedure for the linear and non-linear separation of signals with non-uniform, symmetrical probability distributions, based on both simulated annealing and competitive learning methods by means of a neural network, considering the properties of the vectorial spaces of signals, and using a multiple linearization in the mixture space.

1 Introduction

problem of linear blind separation of sources involves obtaining the signals generated by p sources $\mathbf{x}(t) = [x_1(t),...,x_p(t)]$ from the linear mixture signals, ${\bf e}({\bf t})$ =[e₁(t),...,e_p(t)]¹. The mixture is characterized by an unknown matrix ${\bf A}({\bf t})$ such that:

$$
e(t) = A(t) \mathbf{x}(t) \tag{1}
$$

If the mixture is stationary, then $A(t) = A$. The goal is to estimate $A(t)$ by means of another matrix $\mathbf{W}(t)$ such that the output vector, $\mathbf{s}(t)$ is obtained as follows:

$$
s(t) = W^{-1}(t) e(t)
$$
 (2)

 $\frac{1}{1}$
coinc The output s coincides with the sources, **x**, except for a scale factor and a permutation:

$$
W(t) = A(t) \ P \ D \tag{3}
$$

vertices of the hyperparallelepiped that contains the observation space [7], and for nonwhere **P** is any permutation matrix and **D** is any full-rank diagonal matrix. Many kinds of approaches have been presented concerning the blind separation of sources, with applications in real problems such as communications, pattern recognition, data visualization, speech processing and biomedical signal analysis (EEG, MEG, fMRI, etc), considering the hypothesis that the medium where the sources have been mixed is linear, convolutive or non-linear. There are a great number of solutions for blind separation of sources in several areas [1,2,6,7]. From geometric considerations, and for linear mixtures, various algorithms have been presented, all of which find a matrix that is similar to ${\bf A}$ by determining the slopes ofthe edges that are incident on any one of the linear mixtures [5] or for post-nonlinear mixtures [10]. In [8] an adaptive procedure is described 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. The approach presented in this paper combines the geometric properties of the distributions, with the advantages of competitive neural networks, the main idea being the use of a stochastic technique, simulated annealing, in order to provide fast initial convergence.

2 Proposed method

We propose a method for blind separation of sources that combines adaptive processing with a simulated annealing technique, and which is applied by normalizing the observed space, $e(t)$, in a set of concentric p-spheres in order to adaptively compute the slopes corresponding to the independent axes of the mixture distributions by means of an array of symmetrically distributed neurons in each dimension (Figure 1). A preprocessing stage to normalize the observed space is followed by the learning of the neurons, which estimate the high density regions in a way similar, but not identical to that of self organizing maps [3]. A stochastic method provides a fast initial movement of the weights towards the independent components by generating random values of the weights and minimizing an energy function, being this a way of improving the performance to speed up the convergence of the algorithm. In general, the observation space $(e_1, ..., e_n)$ is subsequently quantized in n spheres of dimension p (p-spheres), circles if $p=2$, each with a radius ρ_k (k=1...n) covering the points as follows:

$$
\rho_{k-1} < \|e(t)\| < \rho_k \qquad \rho_0 = 0 \qquad \forall k \in \{1, \dots, n\} \tag{4}
$$

The number of p-spheres, n, provides the accuracy in the estimation of the independent components. We use $e(\rho_k, t)$ to denote the vector $e(t)$ that verifies (5).

2.1 Competitive Learning

The preprocessing is used to apply a competitive learning technique by means of a neural network whose weights are initially located on the Cartesian edges of the p-dimensional space such that the network has $2p$ neurons, and each neuron w_i being identified with p scalar weights $(w_{i1}, w_{i2},..., w_{ip})$ per p-sphere. The distance between a point, $e(\rho_k, t)$, and the 2p neurons existing in the p-dimensional space (Figure 1) is:

$$
d(i, \rho_k) = || w_i(\rho_k, t) - e(\rho_k, t) || i \in \{1, ..., 2p\} \quad k \in \{1, ..., n\}
$$
\n(5)

A neuron, labelled i^{*}, in a p-sphere ρ_k , is at a minimum distance from $e(\rho_k, t)$:

$$
d(i^*, \rho_k) = \min\{d(i, \rho_k)\} \quad i^* \subseteq i \in \{1, \dots, 2p\} \quad k \in \{1, \dots, n\}
$$
 (6)

The main process for competitive learning when a neuron approaches the density region, in a sphere ρ_k at time t, is given by a learning procedure that activates all the neurons:

$$
\mathbf{w}_i(\rho_k, t+1) = \mathbf{w}_i(\rho_k, t) + \alpha(\rho_k, t) \operatorname{sgn}\left[e(\rho_k, t) - \mathbf{w}_i(\rho_k, t)\right] K_i(t)
$$
\n
$$
K_i(t) = \exp\left(-\eta^{-1}(t) \|\mathbf{w}_i(\rho_k, t) - \mathbf{w}_i \cdot (\rho_k, t)\|^2\right) \quad i^* \subseteq i \in \{1, \dots, 2p\}
$$
\n
$$
(7)
$$

with $\alpha(t)$ and $\eta(t)$ being decreasing learning rates. After the learning process, the weights are maintained in their respective p-spheres, ρ_k , as follows:

$$
\mathbf{w}_i(\mathbf{p}_k, t) = \frac{\mathbf{w}_i(\mathbf{p}_k, t) \mathbf{p}_k}{\|\mathbf{w}_i(\mathbf{p}_k, t)\|} \qquad i \in \{1, \dots, 2p\} \qquad k \in \{1, \dots, n\}
$$
 (8)

A matrix, W , similar to A , is needed (3), and a recursive neural network will uses a continuous function of the scalar weights, as shown in equations (11) and (21). A set, Ω , of matrices can be defined as follows:

$$
\Omega = \left\{ W_{\rho_1} , \dots , W_{\rho_n} \right\} \tag{9}
$$

where, for p dimensions, the matrices $\mathbf{W} \rho_k$ have the following form:

$$
W_{\rho_k} = \begin{pmatrix} W_{11 \rho_k} & \dots & W_{1p \rho_k} \\ W_{p1 \rho_k} & \dots & W_{p p \rho_k} \end{pmatrix} \qquad k \in \{1, ..., n\}
$$
 (10)

For linear systems or "symmetric" non-linear mixtures, the elements of $W\rho_k$ obtained using competitive learning are considered to be the symmetric slopes, in the segment of p-sphere radius ρ_k , between two consecutive weights, for each dimension j, and finally computed in (11) if the following transformation is carried out:

$$
W_{ij\,p_k}^c(t) = \frac{w_{2j\,j}\,(\rho_k,t) - w_{2j\,j}\,(\rho_{k-1},t)}{w_{2j\,j}\,(\rho_k,t) - w_{2j\,j}\,(\rho_{k-1},t)} \qquad i,j \in \{1,..,p\} \quad k \in \{1,..,n\}
$$
(11)

Equation (11) can be simplified for linear media (n=1, ρ_0 =0).

2.2 Simulated Annealing

Simulated annealing is a stochastic algorithm that represents a fast solution to some combinatorial optimization problems. We first propose the use of stochastic learning, such as simulated annealing, in order to find a fast convergence of the weights around the maximum density points in the observation space e (t). For the problem of blind separation of sources we define an energy, E, related to the four-order statistics of the original p sources, as follows:

$$
E = \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} E_{ij}(t) = \sum_{i=1}^{p-1} \sum_{j=i+1}^{p} \langle sum_{22}^2 (s_i(t) s_j(t)) \rangle
$$
 (12)

where $cum_{22}(s_i(t), s_j(t))$ is the the 2x2 fourth-order cumulant of $s_i(t)$ and $s_j(t)$, i.e.:

$$
cum_{22}(s_i(t), s_j(t)) = \langle s_i^2(t) s_j^2(t) \rangle - \langle s_i(t)^2 \rangle \langle s_j(t)^2 \rangle - 2 \langle s_i(t) s_j(t) \rangle^2 \tag{13}
$$

and $\langle x(t) \rangle$ represents the expectation of x (t). The estimation of that energy can be done using the methods described in [6]. The change in global energy, ΔE , is given by:

$$
\Delta E = E(t+1) - E(t) \tag{14}
$$

If $\triangle E \le 0$ then the process accepts the change. If $\triangle E \ge 0$, the system accepts the change providing $P > r$, where r is a number randomly chosen for P, the Boltzmann distribution given $\triangle E$, computed using the equation:

$$
P = e^{-\frac{\Delta E}{T(t)}} \tag{15}
$$

where $T(t)$ is the positive valued temperature at time t that regulates the search granularity for the system's global minimum. If $\Delta E > 0$ and P < r, then the network returns all weights to their original state. The temperature $T(t)$ is calculated, as:

$$
T(t) = \frac{T_0}{1 + \eta(t)}\tag{16}
$$

where T₀ is the initial temperature. We propose the function $\eta(t)$ in (16) should be $\eta(t)=(1+t)^2$ -1, in order to provide fast convergence. With this process, and using r_{ii} to denote a random number in the range $[0,1]$ a separation matrix is computed as:

$$
W_{ij \rho_k}^s(t) = 2 r_{ij} - 1 \qquad i, j \in \{1, \dots, p\} \quad i \neq j \quad k \in \{1, \dots, n\}
$$
 (17)

The proposed energy function (15) depends on a four-order moment; it has been experimentally corroborated in several simulations as an estimator of statistical independence, obtaining good results by estimating statistics over $N=100$ samples. Although the use of simulated annealing does not guarantee finding the global optima with a few number of samples, it provides a good starting point for the next learning process.

2.3 Competitive Learning with Simulated Annealing

Now, a proposed adaptive rule of the weights is the following:

$$
W_{ij \, \rho_k}(t+1) = W_{ij \, \rho_k}^s(t) \, \beta(t) + W_{ij \, \rho_k}^c(t) \, (1-\beta(t))
$$

$$
i \neq j \in \{1, ..., p\} \quad k \in \{1, ..., n\}
$$
 (18)

where $\beta(t)$ is a decreasing function, similar to (16). The purpose of equation (18) is to provide a fast initial convergence of the weights by means of simulated annealing during the epoch in which the adaptation of the network by competitive learning is still inactive. When the value $\beta(t)$ goes to zero, the contribution of the simulated annealing vanishes, and the more accurate estimation by means of competitive learning begins. The main contribution of simulated annealing is the fast convergence compared to the adaptation rule (7). However, the accuracy of the solution when the temperature, $T(t)$, is low depends mainly on the adaptation rule using competitive learning since the energy in (13) continues to decrease until a global minimum is obtained. The use of different approaches in order to estimate the centers of mass, as standard K-means does, is a common practice in expectation maximisation fitting of Gaussians, but the complexity of this procedure and the lack of knowledge of the centroids allows simulated annealing more adecuate. A measure of the convergence with the number of samples is shown in Figures 3 and 4, which compare competitive learning and simulated annealing, using the root mean square error, $\epsilon(t)$, where the diagonal elements are the unity:

$$
\epsilon(t) = \frac{1}{p(p-1)} \left(\sum_{i \neq j} \left(W_{ij}(t) - A_{ij}(t) \right)^2 \right)^{\frac{1}{2}} \qquad i, j \in \{1, ..., p\}
$$
 (19)

The kurtosis parameter provides information concerning the distribution of, $x(t)$, i.e.,

$$
\sum_{i \neq j} (W_{ij}(t) - A_{ij}(t))^2
$$
 (19)
ateter provides information concerning the distribution of, x(t), i.e.,

$$
k_x = \frac{ - 3 < x(t)^2>^2}{^2}
$$
 (20)
e expectation of x (t). Figure 3 shows the root mean square error for
 $p=2$ signals and $n=1$, with the two sources having kurtosis of κ_{1}

where $\leq x$ (t) is the expectation of x (t). Figure 3 shows the root mean square error for linear mixtures of p=2 signals and n=1, with the two sources having kurtosis of $\kappa_{s1} = 0.02$ and $\kappa_{s2} = 0.02$ respectively in several experiments. Using simulated annealing and 0.02 and $\kappa_{s2} = 0.02$, respectively, in several experiments. Using simulated annealing and 10000 samples the error remains at $\epsilon = 0.05$, whereas using simulated annealing and 10000 samples the error remains at ϵ =0.05, whereas using simulated annealing and
competitive learning the error becomes ϵ =0.01 with the same number of iterations. In competitive learning the error becomes ϵ =0.01 with the same number of iterations. In
Figure 4 the root mean square error in the case of n=3 and n=1 is shown, the sources Figure 4 the root mean square error in the case of $p=3$ and $n=1$ is shown, the sources having kurtosis values of $\kappa_{s1} = 3.1$, $\kappa_{s2} = 3.5$ and $\kappa_{s3} = 3.2$, respectively. With a larger number of sources to be separated, using simulated annealing and 15000 samples the error remains at ϵ =0.06, whereas using simulated annealing and competitive learning the
error becomes ϵ =0.01 error becomes ϵ =0.01.

3 Separation matrix

 $\tilde{\xi}$

We will use expression (11) for computation of the weights, although in the general nonlinear case above expression must be replaced by a similar one (Figure 2), as follows:

$$
W_{ij \, \rho_k}^c(t) = \frac{w_{\xi(j) \, i}(\rho_k, t) - w_{\xi(j) \, i}(\rho_{k-1}, t)}{w_{\xi(j) \, j}(\rho_k, t) - w_{\xi(j) \, j}(\rho_{k-1}, t)} \quad i, j \in \{1..p\} \quad k \in \{1...n\}
$$

(*j*) $\in \{\xi(1) < \xi(2) < ... < \xi(p) \mid d(\xi(j), \rho_k) < d(\xi(m), \rho_k)\} \quad m \in \{1...2p\} \quad m \neq j$ (21)

 $\begin{array}{c}\n a \\
d(\xi) \\
d\n \end{array}$
dim
rovide
k us Equation (21) means that the p-dimensional subspace associated to the neurons labelled $\mathcal{L}(1) = \mathcal{L}(1)$ inclus that the p-unnershonal subspace associated to the neurons fabelled
 $\mathcal{L}(1) = \mathcal{L}(1) = \frac{1}{2}$ $(\xi(1), ..., \xi(p))$ around point \mathbf{e}_p provides the linear contour, between the radius ρ_k and ρ_k . where the mixture can be considered as linear. Then, we come up with this way to recover the sources; the network uses the typical recursive recall, taking into account the p-sphere quantization in the observation space and the matrix computed in (11) or (21) :

$$
s_{i}(t+1) = e_{i}(\rho_{k}, t) - \sum_{j=1}^{p} W_{ij \rho_{k}}(t) s_{j}(t) \quad i \in \{1, ..., p\} \quad i \neq j \quad k \in \{1, ..., n\}
$$
 (22)

 $\frac{1}{f=1}$
used
and
lts
, ct_i, This expression is also used by the simulated annealing process in order to compute the energy function in (12) and (13).

4 Simulation results

The crostalk parameter, ct_i , is used to verify the similarity between x_i and s_{i_n} as follows:

$$
ct_{i} = 10 \log \left(\frac{\sum_{t=1}^{n} (s_{i}(t) - x_{i}(t))^{2}}{\sum_{t=1}^{N} (s_{i}(t))^{2}} \right)
$$

 $i \in \{1,...,p\}$ (23)
on (Figures 5-8) concerns the separation of a mixture of two real signal:
 x_{i} , and the $\mathbf{W}\rho_{1}$ matrix computed after 10000 samples were:

The simulation (Figures 5-8) concerns the separation of a mixture of two real signals. The ${\bf A}$ matrix, and the ${\bf W}\rho_1$ matrix computed after 10000 samples were:

$$
A = \begin{pmatrix} 1 & -0.8 \\ 0.8 & 1 \end{pmatrix} \qquad W_{\rho_1} = \begin{pmatrix} 1 & -0.791 \\ 0.788 & 1 \end{pmatrix}
$$
 (24)

The crosstalk parameter of the separated signals, $s_1(t)$ and $s_2(t)$, was $ct_1(t) = -24$ dB and $ct_2(t) = 23 dB$, respectively. It has been verified that the greater the kurtosis of the signals the more accurate and faster is the estimation, except for the case in which the signals are not well conditioned or are affected by noise.

5 Conclusions

We have shown a new powerful adaptive-geometric method based on competitive unsupervised learning and simulated annealing, in order to find the distribution axes of the observed signals or independent components, by means of a piecewise linearization in the mixture space, the use of simulated annealing in optimisation of a four-order statistical criterion being an experimental advance. The use of new metaheuristics we propose has benefits such as robustness against local minima, the parallel search for various solutions, a high degree of flexibility in the evaluation function and provide a good starting point. The algorithm, in its current form, presents some drawbacks concerning the application of simulated annealing to a high number of signals, and the complexity of the procedure $O(2^p p^2 n)$ for the separation of nonlinear mixtures.

Acknowledgements

This work has been supported in part by the Spanish CICYT project TIC98-0982.

References

[1] Bell A.J., and Sejnowski TJ. An information-maximization approach to blind separation and blind deconvolution. Neural Computation 7(6), 1995, 1129-1159.

[2] Girolami M. The Latent Variable Data Model for Exploratory Data Analysis and Visualization: A Generalisation of the Nonlinear Infomax Algorithm. Neural Processing Letters 8 (1), 1998, 27-39.

[3] Haykin S. Neural Networks. Prentice Hall, 1999.

[4] Jutten C. and Herault J. Blind Separation of Sources, Part I: an Adaptive Algorithm Based on Neuromimetic Structure. Signal Processing 24 (1), 1991, 1-10.

[5] Lin J.K., and Cowan J.D. Faithful Representation of Separable Input Distributions. Neural Computation 9, 1997, 1305-1320.

[6] Mansour A, Kardec Barros A, Ohnishi N. Comparison among three estimators for high order statistics". Fifth International Conference on Neural Information Processing (ICONIP'98), Kitakyushu (Japan), 21-23 October, 1998, 899-902.

[7] Puntonet C.G., and Prieto A. Neural Net Approach for Blind Separation of Sources based on Geometric Properties. Neurocomputing, 18(3), 1998, 141-164.

[8] Puntonet C.G., Alvarez M.R., Prieto A., Prieto B. Separation of Speech Signals for Nonlinear Mixtures. Lecture Notes in Computer Science, 1607, Springer 1999, 665-673. [9] Simpson P.K. Artificial Neural Systems. Pergamon Press, 1991.

[10] A. Taleb, C. Jutten. Nonlinear source separation: The Post-Nonlinear Mixtures. In Proceedings of ESANN'97, D Facto Brussels, Belgium, 1997, 279-284.

Figure 1. Concentric spheres

Figure 2. Subspaces for non-linear separation

Figure 3. RMS error comparison for $p=2$

 $(t=2000)$

 $_{\rm e2}$.
w12

Figure 5. Mixture space of two real signals

Figure 6. Weights with S.A. Figure 7. Weights with C.L $(t=10000)$

Figure 8. Sources, mixtures and separated signals