ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 337-342

Lattice ICA for the Separation of Speech Signals.

Manuel Rodríguez-Álvarez [1], Fernando Rojas[1], Elmar W. Lang[2],
Ignacio Rojas[1], Carlos G. Puntonet[1], M. Salmerón[1]

[1] Departament of Architecture and Computer Technology. University of Granada (Spain)
{mrodriguez, frojas, irojas, cgpuntonet}@atc.ugr.es
[2] Institute of Biophysics, University of Regensburg Germany)
elmar.lang@biologie.uni-regensburg.de

Abstract. This work explains a method for blind separation of a linear mixture of sources, through geometrical considerations concerning the scatter plot. This method is applied to a mixture of several sources and it obtains the estimated coefficients of the unknown mixture matrix A and separates the unknown sources.

1. Introduction

When p different source signals propagating through a real medium have to be captured by sensors, these sensors are sensitive to all sources $s_i(t)$ and thus the signal $x_k(t)$, observed at the output of sensor k, is a mixture of source signals. With a linear and stationary mixing medium the sensor signals can be described by:

$$\vec{x}(t) = A\,\vec{s}(t) \qquad \textbf{(1)}$$

where $\vec{x}(t) = (x_1(t),\ \ldots,\ x_n(t))^T$ is an experimentally observable $(n \times 1)$-sensor signal vector s(t), with $\vec{s}(t) = (s_1(t),\ \ldots,\ s_p(t))^T$ is a $(p \times 1)$ - unknown source signal vector having stochastic independent and zero-mean non-Gaussian elements $s_i(t)$, and A is a $(n \times p)$ unknown full-rank and non-singular mixing matrix. The solution of the blind signal separation (BSS) problem consists of retrieving the unknown sources $s_i(t)$ from just the observations. To achieve this it is necessary to apply the hypotheses that the sources $s_i(t)$ and the mixture matrix A are unknown, that the number n of sensors is at least equal to the number p of sources, i.e. $n \geq p$, and that the components of the source vector are statistically independent yielding:

$$p(\vec{s}) = \prod_{i=1}^{n} p(s_i) \qquad \textbf{(2)}$$

In order to solve the BSS problem a separating matrix W is computed whose output is an estimate of the vector $\vec{s}(t)$ of the source signals such that:

$$\vec{y}(t) = W^{-1}\,\vec{x}(t) \qquad \textbf{(3)}$$

Any BSS algorithm can only obtain W subject to:

$$W^{-1}A = DP \qquad \textbf{(4)}$$

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 337-342

with a diagonal scaling matrix *D* modified by a permutation matrix *P*. Recently, BSS and ICA (Independent Component Analysis) have received much attention because of its potential applications in signal processing. A great diversity of estimation methods have been proposed based on some kind of statistical analysis, neural networks [7], the entropy concept [3], the geometric structure of the signal spaces [1], [6], the fixed-point algorithm FastICA [5], the maximum likelihood stochastic gradient algorithm [2], the Jade algorithm [4], among others.

2. Principles of the new method

For $p = 2$ and with bounded values in a uniform distribution, the observed signals $(x_1(t),\ x_2(t))$ form a parallelogram in the $(\vec{x}_1, \vec{x}_2)$ space, as shown in Figure 1. We have demonstrated [8] that, through a matrix transformation, the coefficients of the matrix coincide with the slopes of the parallelogram. It can be seen that for random uniform sources, the parallelogram representing the scatter plot $(\vec{x}_1, \vec{x}_2)$ is geometrically bounded within the segments between the points P_1 to P_4. The slopes of these segments give the coefficients of the estimated mixture matrix W. In order to obtain these segments, it is necessary to estimate the coordinates of those points P_i, $i = 1, 2, 3, 4$. Assuming non-uniformly distributed signal as the sources, for example speech signals with an underlying super-Gaussian distribution; the form of the sensor signal distribution in the scatter plot is highly non-uniform too. In this case it is not sufficient to estimate the borders of the bounded scatter plot. Rather, it is necessary to detect the directions of high density in the scatter plot. These directions are called ICA axes (ICA-1 ; ICA-2).

2.1 Description of the algorithm

First of all, the algorithm computes the kurtosis of each component of the sensor signals and also the correlation coefficients between all observations. This is to detect whether the underlying source signal distributions correspond to sub- or super-Gaussian distributions. According to the Central Limit Theorem, mixtures will tend to be closer to Gaussian than the original ones. Consequently, kurtoses of the mixtures will be closer to zero (Gaussian distribution) than the sources:

$$\left|Kurt(x_i)\right| \leq \max\left\{ \left|Kurt(s_j)\right| \right\} \ ;\ i, j \in [1, ..., n] \qquad \textbf{(5)}$$

In any case, for mixtures of two signals, they will tend to preserve the sub- or super-Gaussian nature of the original signals, assuming that both sources have the same sign in the kurtosis. If the kurtoses of all observations are positive, the algorithm searches for high density regions of the sensor signal distribution. With sub-Gaussian signals, the algorithm estimates the bounding box of the parallelogram representing the scatter plot.

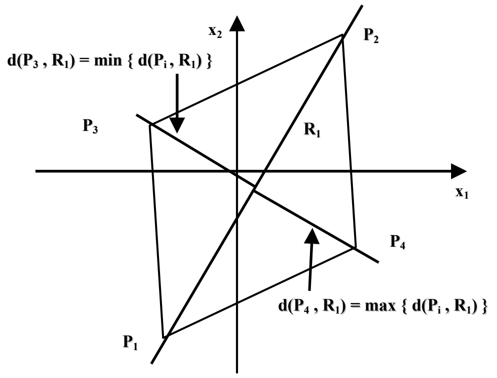

Figure 1. Scatter plot: Representative points and ICA axes.

The algorithm subdivides the scatter plot $(\vec{x}_1, \vec{x}_2)$ into a regular lattice of cells with N-rows and M-columns. Then, the algorithm computes the number of cells in the lattice in which the number of points inside it is greater than a given threshold TH. The distribution of sensor signals within each of these cells then is replaced by a prototype sensor signal vector. The prototype vector mostly does not point towards the centre of the cell because its position is weighted by the density of points (x_{1i}, x_{2i}) in this cell. The next step of the algorithm finds those points which either form the border of the hyperparallelepiped or mark the high density regions of the sensor signal distribution in the space, by looking for cells that have an empty neighborhood (such cells have fewer points than the threshold TH). Then these cells without a complete neighborhood form the border of the distribution encompassing NR data points in the scatter plot. The algorithm then computes the coordinates of $P_1 = (p_{11}, p_{12})$ and $P_2 = (p_{21}, p_{22})$. The scatter plot has been reduced to NR data points which, in two dimensions, represent pairs of coordinates (x_{1i}, x_{2i}). In this reduced set of NR data points, there exist data points P_1 and P_2 with largest Euclidean distance between them in the scatter plot :

$$d(P_1, P_2) = \max_{i,j \in (1,2,....NR)} d(P_i, P_j) \qquad \textbf{(6)}$$

Once points P_1 and P_2 have been identified, the algorithm calculates the equation of the straight line R_1 which passes through these points P_1 and P_2 :

$$Ax_1 + Bx_2 + C = 0 \quad ; \quad \text{being} \qquad \textbf{(7)}$$

$$A = (p_{22} - p_{12}), \quad B = (p_{11} - p_{21}), \quad C = (p_{21} - p_{12}) - (p_{22} - p_{11}) \qquad \textbf{(8)}$$

Next, the algorithm estimates the coordinates of the points $P_3 = (p_{31}, p_{32})$ and $P_4 = (p_{41}, p_{42})$ as follows: the straight line R_1 divides the scatter plot $(\vec{x}_1, \vec{x}_2)$ into two subspaces, being R_1 the border between them. Data points which lie within one of these subspaces yield a nonzero result in Eq. (7). For example, data points lying above the straight line R_1 yield a negative result in Eq. (7). There is then one data point $P_3 = (p_{31}, p_{32})$ which provides the most negative value of all possible outcomes of Eq. (7), hence which also represents the point with the greatest Euclidean distance from the straight line R_1 in the subspace above R_1. In the same way, points in the other subspace, below the straight line R_1, yield a positive result in Eq. (7). Again, there is one point $P_4 = (p_{41}, p_{42})$ that provides the most positive value of all possible results from Eq. (7), and which is also the point with greatest Euclidean distance from the straight line R_1 in the subspace below R_1.

Once the characteristic points of the parallelogram have been obtained, the algorithm computes either the slopes of the segments ($\overline{P_1P_3}$ and $\overline{P_1P_4}$ or, equivalently $\overline{P_2P_4}$ and $\overline{P_3P_2}$) in case of sub-Gaussian densities or the slopes of the diagonals ($\overline{P_1P_2}$ and $\overline{P_3P_4}$) in case of super-Gaussian densities in order to obtain the slopes of the ICA axes and the coefficients of the matrix W as in Eq. (9) (see Figure 1):

$$\left(\frac{a_{12}}{a_{22}}\right)^{-1} = \frac{p_{32} - p_{12}}{p_{31} - p_{11}} \quad ; \quad \left(\frac{a_{21}}{a_{11}}\right) = \frac{p_{42} - p_{12}}{p_{41} - p_{11}} \qquad \textbf{(9)}$$

Using the coefficients of matrix W, the algorithm computes the inverse matrix W^{-1} and reconstructs the unknown source signals $\vec{s}(t)$ (see Eq. (3)).

2.2 Further enhancements

The computational order of the algorithm is polynomial:

$$Comput - Order = (DataPoints^2 \cdot XColumns \cdot YRows) \qquad \textbf{(10)}$$

As a further improvement, we propose the reduction of the number of points at the beginning of the algorithm with a random elimination through all the space of the joint distribution of the mixtures as long as enough data points are kept to correctly estimate the sources. A more elaborated proposal is eliminating those points of the joint distribution of the mixtures which lay within a calculated radius near the center of the joint distribution, because they are useless for the algorithm, due to its nature of computing contours using points whose Euclidean distances are the highest. From experimental results, we have derived equations (11) and (12) for the calculation of the radius based on the kurtosis and correlation of the mixture signals.

For sub-Gaussian mixtures, the algorithm will try to find the contour of the sensor signal distribution. In this case we determine the exclusion radius as follows :

$$R = \frac{\alpha}{\rho(x)^2 + 0.1} \cdot \overline{x} \qquad \textbf{(11)}$$

where α is a constant (experimentally, a value of α=7.5 was applied), $\rho(x)$ is the correlation of the mixtures and

$$\overline{x} = \sqrt{\sum_{j=1}^{N} x(1, j)^2 + x(2, j)^2} \qquad \textbf{(12)}$$

For super-Gaussian mixtures (positive kurtosis), the algorithm will search for high density regions of the scatter plot. Thus, the exclusion radius was calculated as:

$$R = 1.5 \cdot \bar{x} \qquad \textbf{(13)}$$

3. Simulations and Results

The new algorithm, named as "LatticeICA", has been tested on various ensembles of artificial sensor signals with an arbitrary number of samples drawn at random from sub- and super-Gaussian distributions, as well as with real world speech signals. To quantify the performance achieved we calculate both a crosstalking error *E (W,A)* of the original and recovered source signals as proposed by Amari et al. [2] as well as a component wise crosstalk.

Application to separate speech signals.

In this simulation the algorithm separate two super-Gaussian speech voice signals with 33684 samples each. The lattice was automatically computed to be 13 rows and 13 columns, using TH = 149. The original and estimated matrices were:

$$A = \begin{bmatrix} 1 & 0.70 \\ -0.30 & 1 \end{bmatrix}; \; W = \begin{bmatrix} 1 & 0.72 \\ -0.27 & 1 \end{bmatrix} \qquad \textbf{(14)}$$

The joint distribution of the mixtures points out the super-Gaussian nature of the sources (see Figure 2). The matrix performance index for this simulation was $E\,(W, A) = 0.104$, with Crosstalk1 (E_{s1}) = -35 dB and Crosstalk2 (E_{s2}) = -32 dB. In Figure 2 it is shown how the algorithm searches for the lines of higher density instead of the contour plot.

4. Conclusions

In this work, a new geometry-based method for blind separation of sources has been developed, which greatly reduces the complexity and computational load inherent in the standard geometrical ICA algorithms. This new algorithm is based on a tessellation of the input space where in each cell a code book vector is determined to represent the center of gravity of the local distribution of sample vectors. Depending on the type of distribution, the slopes of the border lines or the diagonals are determined to obtain the coefficients of the estimated mixing matrix *W*. The method lends itself for an easy hardware implementation and is also very intuitive. Furthermore, this method could be used to detect the perimeter or outlines in simple two-dimensional figures. In the future we will intend to implement this method for more than two signals working in the p-dimensional space.

Acknowledgement

This work has been supported by the Spanish CICYT Projects TIC2001-2845 "PROBIOCOM - Procedures for Biomedical and Communications for BSS" and TIC2000-1348 "Hybrid procedures for parallel optimization in clusters".

ESANN'2004 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 28-30 April 2004, d-side publi., ISBN 2-930307-04-8, pp. 337-342

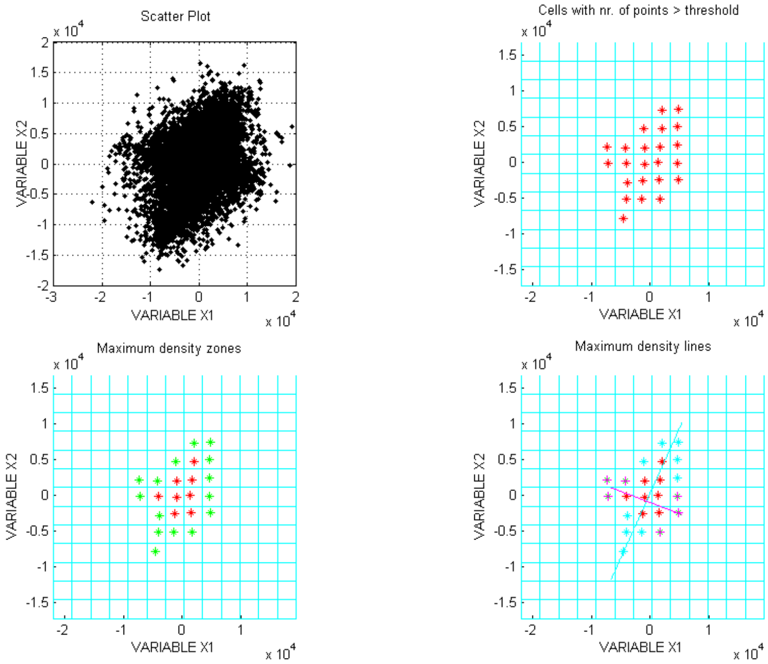

Figure 2. Performance of the LatticeICA algorithm for a two real voice signals mixture.

References

1. M. R. Álvarez, C. G. Puntonet, I. Rojas: Separation of Sources based on the Partitioning of the Space of Observations. LNCS, Vol. 2085, 762 – 769 (2001).
2. S. I. Amari, A. Cichocki, H. H. Yang: A new learning algorithm for blind signal separation. Proceedings of NIPS'96, 757 – 763 (1996).
3. A. J. Bell, T. J. Sejnowski: An information-maximisation approach to blind separation and blind deconvolution. Neural Computation, 7, 1129-1159 (1995).
4. J. F. Cardoso: High-order contrasts for Independent Component Analysis. Neural Computation, 11, 157-192 (1999).
5. A. Hyvärinen, J. Karhunen, E. Oja.: Independent Component Analysis. John Wiley & Sons, New York (2001).
6. A. Jung, F. J. Theis, C. G. Puntonet, E. W. Lang: FASTGEO: A histogram based approach to linear geometric ICA. Proceedings of ICA'01, 349-354 (2001).
7. C. Jutten, J. Hérault, P. Comon, E. Sorouchiary: Blind separation of sources, Parts I, II, III. Signal Processing, 24, nº 1, 1-29 (1991).
8. C. G. Puntonet, A. Prieto: Neural net approach for blind separation of sources based on geometric properties. Neurocomputing, 18, 141-164 (1998).