Solving Independent Component Analysis Contrast Functions with Particle Swarm Optimization

Jorge Igual[1], Jehad Ababneh[2], Raul Llenares[1], Julio Miro-Borras[1], and Vicente Zarzoso[3]

[1] Universidad Politecnica de Valencia, Camino de Vera s/n, 46022 Valencia, Spain
[2] Jordan University of Science and Technology 22110 Irbid, Jordan
[3] Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis 06903 Sophia Antipolis Cedex, France
jigual@dcom.upv.es

Abstract. Independent Component Analysis (ICA) is a statistical computation method that transforms a random vector in another one whose components are independent. Because the marginal distributions are usually unknown, the final problem is reduced to an optimization of a contrast function, a function that measures the independence of the components. In this paper, the stochastic global Particle Swarm Optimization (PSO) algorithm is used to solve the optimization problem. The PSO is used to separate some selected benchmarks signals based on two different contrast functions. The results obtained using the PSO are compared with classical ICA algorithms. It is shown that the PSO is a more powerful and robust technique and capable of finding the original signals or sources when classical ICA algorithms give poor results or fail to converge.

Keywords: Independent Component Analysis, Particle Swarm Optimization.

1 Introduction

Independent Component Analysis (ICA) and its most popular application Blind Source Separation (BSS) is a very active research area. The theoretical problem has been clearly stated since a long time [1]. In this paper we will concentrate in the linear noiseless instantaneous model $\mathbf{x} = \mathbf{A}\mathbf{s}$ where $\mathbf{x}$ is the observed vector that is a linear transformation (matrix $\mathbf{A}$) of a random vector $\mathbf{s}$ whose components are statistically independent. In BSS terminology, $\mathbf{x}$ is the mixture vector, $\mathbf{A}$ is the mixing matrix and $\mathbf{s}$ is the source vector. Many ICA algorithms can be decomposed into two steps; in the first one, the second order statistics are exploited, imposing the decorrelation of the signals (whitening step). The second step consists in the estimation of an orthogonal matrix that imposes the independence, being necessary the use of higher order statistics. In matrix notation,

$$\mathbf{y} = \mathbf{B}\mathbf{x} = \mathbf{U}\mathbf{W}\mathbf{x} \tag{1}$$

K. Diamantaras, W. Duch, L.S. Iliadis (Eds.): ICANN 2010, Part II, LNCS 6353, pp. 519–524, 2010.

where $\mathbf{W}$ is the whitening matrix and $\mathbf{U}$ is the orthogonal one. The whitened vector is expressed as $\mathbf{z} = \mathbf{WAs}$, with $E[\mathbf{zz}^T] = \mathbf{I}$, and the uncorrelated unit variance constraint $E[\mathbf{yy}^T] = \mathbf{I}$, being $\mathbf{y}$ the recovered or estimated sources.

The independence hypothesis must be approximated and, as a consequence, the estimation of the sources is transformed in an optimization problem defined by a contrast (cost) function that is minimum when the estimated sources are as independent as possible, i.e., the matrix $\mathbf{BA}$ equals the product of a permutation (order indeterminacy) and a diagonal matrix (scale indeterminacy). We present in this paper the Particle Swarm Optimization PSO approach to solve the ICA problem. The use of the PSO to solve the ICA problem is proposed to overcome the trapped-in-local-optimum problems of gradient descent traditional approaches. Although PSO and ICA are very active research areas, there are few works about the application of PSO in ICA; see, e.g., [5], [6], and they are focused in some particular part of the ICA solution, not in a general analysis of cost functions.

2 Contrast Functions in ICA

2.1 Mutual Information

The Mutual Information MI is defined as the Kullback-Leibler divergence or relative entropy between the joint density and the product of the marginal distributions; it is non negative and equals to zero only if the distributions are the same, i.e., MI is a contrast function for ICA:

$$MI(\mathbf{y}) = KL(\mathbf{y}; \prod_i p(y_i)) = \int p(\mathbf{y}) \log \frac{p(\mathbf{y})}{\prod_i p(y_i)} d\mathbf{y} \tag{2}$$

It is related to the differential entropy, $MI(\mathbf{y}) = \sum_i H(y_i) - H(\mathbf{y})$, where the differential entropy of a random variable u is defined as $H(u) = -\int p(u) \log p(u) du$, that can be seen as a measure of the randomness of the variable u. Using $H(\mathbf{y}) = H(\mathbf{x}) + \log|\det \mathbf{B}|$, the contrast, up to an additive constant term, can be expressed as:

$$MI(\mathbf{y}) = \sum_i H(y_i) - \log|\det \mathbf{B}| \tag{3}$$

with the advantage that only one dimensional distributions are involved instead of multidimensional densities. In the case where the observations are first whitened, i.e., $\mathbf{y} = \mathbf{Uz}$, $E[\mathbf{zz}^T] = \mathbf{I}$, we only have to estimate the remaining orthogonal matrix, and the contrast is reduced to the sum of the marginal entropies of $\mathbf{y}$:

$$MI(\mathbf{y}) = \sum_i H(y_i) \quad , \; E[\mathbf{yy}^T] = \mathbf{I} \tag{4}$$

As a conclusion, ICA can be interpreted as a minimum entropy method under the whitening assumption. Hence, the MI contrast is equivalent to find the marginal distributions as far as possible from Gaussianity.

2.2 Higher Order Cumulants

The cross-cumulants are equal to zero for independent variables. In ICA, it means that $C_{ij}(\mathbf{s}) = \sigma_i^2 \delta_{ij}$, $C_{ijkl}(\mathbf{s}) = k_i \delta_{ijkl}$ with $\delta_{ij} = 1$ for $i = j$ and $\delta_{ij} = 0$ otherwise, $\delta_{ijkl} = 1$ for $i = j = k = l$ and $\delta_{ijkl} = 0$ otherwise, σ_i^2 is the variance and k_i is the kurtosis of the source component s_i , i.e., $k_i = E[s_i^4] - 3E^2[s_i^2]$. Then, if there is no prior knowledge about the kurtosis, the contrast function J_{cum} is:

$$J_{cum} = -\sum_i C_{iiii}^2(\mathbf{y}) \tag{5}$$

This is equivalent up to a constant to $\sum_{ijkl \neq iiii} C_{ijkl}^2(\mathbf{y})$ since $E[\mathbf{y}\mathbf{y}^T] = \mathbf{I}$. JADE algorithm [4] approximates the independence by minimizing a smaller number of cross cumulants, $\phi_{JADE} = \sum_{ijkl \neq ijkk} C_{ijkl}^2(\mathbf{y})$.

3 Particle Swarm Optimization for ICA

PSO is a stochastic optimization technique that was first introduced in 1995 by Eberhart and Kennedy [5]. It is a global optimization algorithm that simulates the swarming behavior of birds, bees, fish, etc. The PSO has a comparable performance to other stochastic optimization technique like genetic algorithm (GA) and simulated annealing. A major advantage of the PSO is its ease of implementation in both the context of coding and parameter selection.

The PSO starts with an initial population of individuals (to be termed swarm of particles). Each individual (particle) in the swarm is randomly assigned an initial position and velocity within the solution space. The position of the particle is an N-dimensional vector that represents a possible set of the unknown parameters to be optimized. Each particle in the swarm starts from its initial position at its initial velocity in order to find the position with global minimum (or maximum). During the algorithm search, the velocity and position of each particle is updated based on the individual and the swarm experience according to:

$$\begin{gathered} v_{mn}^t = v_{mn}^{t-1} + U_{n1}^t(0,\varphi_1)(pbest_{mn}^t - x_{mn}^{t-1}) + U_{n2}^t(0,\varphi_2)(gbest_{mn}^t - x_{mn}^{t-1}) \\ x_{mn}^t = x_{mn}^{t-1} + \Delta t\, v_{mn}^t \end{gathered} \tag{6}$$

where v_{mn} and x_{mn} represents the velocity and position of the *m-th* particle in the nth dimension, respectively. The superscripts *t* and *t-1* denote the time index of the current and the previous iterations, $U^{t}_{n1}(0,\varphi_1)$ and $U^{t}_{n2}(0,\varphi_2)$ are two different, uniformly distributed random numbers in the intervals [0, φ_1] and [0, φ_2], respectively . These random numbers are generated at each iteration and for each particle. The first term in (6) indicates that particle's current velocity depends on its previous velocity while the other two terms represent the effect of the individual (particle's best position (pbest)) and the swarm experience (neighborhood best position (gbest)) on the behavior of the particle. Δt represents a given time step (usually chosen to be one). The goodness of the new particle position (possible solution) is measured by evaluating a suitable contrast or fitness function.

In this article, Clerc's constriction method is used [6]. It consists on a strategy for the placement of constriction coefficient (χ) by which the whole equation (6) is multiplied. This coefficient is used to control the convergence of the particle, prevent explosion and ensure convergence.

$$\chi = \frac{2}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}} \tag{7}$$

where $\varphi = \varphi_1 + \varphi_2 > 4$. The parameter φ is commonly set to 4.1 and $\varphi_1 = \varphi_2$ which result in an approximate value of 0.7298 for χ .

4 Results and Discussion

In this section, the use of Clerc's constriction PSO method to solve the optimization problem in ICA is demonstrated and different simulation examples are presented. In our implementation of the PSO algorithm, a swarm size of 25 particles, 500 iterations, $\varphi_1 = \varphi_2 = 2.05$, $\chi = 0.7298$ and circular population topology were used. The results obtained are compared with classical ICA algorithm FastICA [7]. In order to test the performance of the algorithm with different but standard signals, we use the benchmarks proposed in the ICALAB package [8]. The ICA contrasts analyzed are the sum of the marginal entropy and the fourth order cumulant (kurtosis), corresponding to equations (4) and (5), respectively. It is worth mentioning that the elements of the PSO output vector are reshaped to form the demixing matrix B in equation (1), which is then normalized and made orthogonal before used in the contrast function in order to impose these constraints on the demixing matrix. The PSO output is an N-dimensional vector (position of the particle with best fitness) that represents a possible set of all the elements of the matrix **B** in equation (1). The reshaping is necessary to transfer the vector into a square matrix.

In the first example, 10 sparse (smooth bell-shape) sources that are approximately independent are randomly mixed. Some ICA algorithms have failed to separate such sources, so we want to test if PSO can overcome these difficulties. The corresponding recovered signals using the PSO and the two different contrast functions are shown in the Figures 1 and 2. The recovered signals are in a very good agreement with the

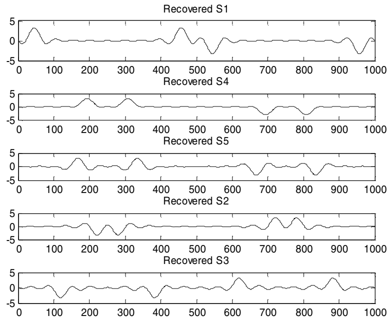

Fig. 1. The recovered signals of Example 1 using PSO and the entropy contrast function

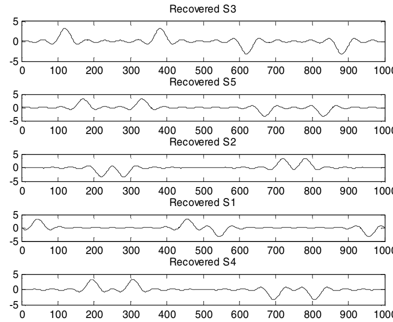

Fig. 2. The recovered signals of Example 1 using PSO and the cumulant contrast function

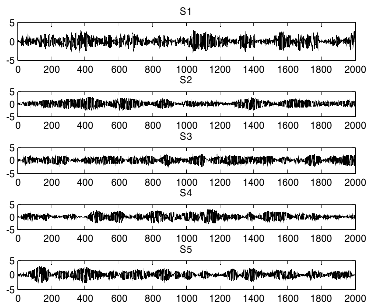

Fig. 3. The original signals of Example 2

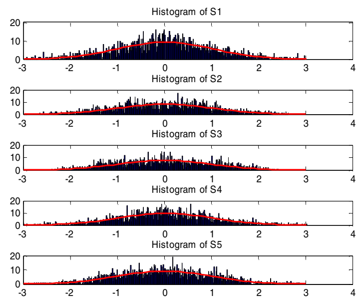

Fig. 4. The Histogram of the signals in Example 2

original signals (we do not show them for the lack of space) with a different ordering and some of them with a change of sign. The recovered signals using FastICA algorithm are also in comparable agreement with the original signals.

Five fourth order colored sources with a distribution close to Gaussian belonging to the same ICALAB package were used in the second example. This is the most challenging case, since the signals are close to Gaussian distribution and ICA algorithms can estimate at most one Gaussian component. Figure 3 shows the original signals of Example 2.

The histograms of the sources are shown in Figure 4 to indicate proximity to Gaussianity. The recovered signals of Example 2 using the PSO with the kurtosis contrast function are shown in Figure 5.

Comparing the signals in Figure 3 and 5, it can be seen that the proposed methodology was successful in recovering some of the original signals with acceptable quality and the other signals with poor quality. On the other hand, the FastICA algorithm has completely failed to separate the mixed signals of this example. The algorithm in the ICALAB package only outputs one signal and does not give any output for the remaining signals. In addition, the algorithm outputs the statement “Too many failures to converge. Giving up”. This example shows the robustness of PSO compared to

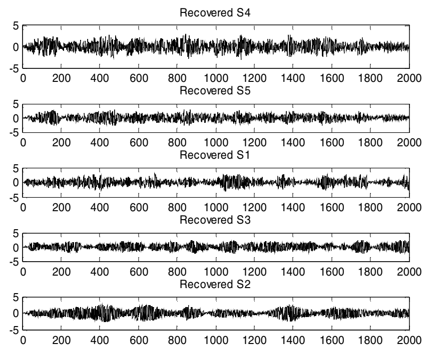

Fig. 5. The recovered signals of Example 2 using PSO and the kurtosis contrast function

classical optimization techniques. In order to assure this result, other ICA algorithms based on other optimization paradigms (gradient based and versions of it) was tested and confirmed this result, which is in agreement with the comments included in the ICALAB toolbox "This is a rather difficult benchmark".

Acknowledgments. This work has been partially funded by the Valencia Regional Government (Generalitat Valenciana) through project GV/2010/002 (Conselleria d'Educacio) and by the Universidad Politecnica de Valencia under grant no. PAID-06-09-003-382.

References

[1] Cardoso, J.F.: Blind Signal Separation: statistical principles. Proceedings of the IEEE 86(10), 2009–2025 (1998)
[2] Krusienski, D.J., Jenkins, W.K.: Nonparametric density estimation based ICA via PSO. In: Proc. ICASSP 2005, vol. IV, pp. 357–360 (2005)
[3] Liu, C.C., Sun, T.Y., Li, K.Y., Hsieh, S.T., Tsai, S.J.: Blind Source Separation using cluster PSO technique. In: Proc. 25th IAESTED Artificial Intelligence and Applications, pp. 289–294 (2007)
[4] Cardoso, J.F., Souloumiac, A.: Blind beamforming for non Gaussian signals. Proc. Inst. Elec. Eng. 140(6), 395–401 (1993)
[5] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)
[6] Clerc, M., Kennedy, J.: The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. 6, 58–73 (2002)
[7] Hyvarinen, A., Oja, E.: A fast fixed point algorithm for independent component analysis. Neural Computation 9(7), 1483–1492 (1997)
[8] Cichocki, A., Amari, S., Siwek, K., Tanaka, T., Phan, A.H., Zdunek, R.: ICALAB – MATLAB Toolbox Ver. 3 for signal processing