Hybridizing Genetic Algorithms with ICA in Higher Dimension

Juan Manuel Górriz[1], Carlos G. Puntonet[2],
Moisés Salmerón[2], and Fernando Rojas Ruiz[2]

[1] E.P.S. Algeciras, Universidad de Cádiz
Avda. Ramón Puyol s/n, 11202 Algeciras Cádiz, Spain
juanmanuel.gorriz@uca.de
[2] E.S.I., Informática, Universidad de Granada
C/ Periodista Daniel Saucedo, 18071 Granada, Spain
{carlos,moises}@atc.ugr.es

Abstract. In this paper we present a novel method for blindly separating unobservable independent component signals from their linear mixtures, using genetic algorithms (GA) to minimize the nonconvex and nonlinear cost functions. This approach is very useful in many fields such as forecasting indexes in financial stock markets where the search for independent components is the major task to include exogenous information into the learning machine. The GA presented in this work is able to extract independent components with faster rate than the previous independent component analysis algorithms based on Higher Order Statistics (HOS) as input space dimension increases showing significant accuracy and robustness.

1 Introduction

The starting point in the Independent Component Analysis (ICA) research can be found in [1] where a principle of redundancy reduction as a coding strategy in neurons was suggested, i.e. each neural unit was supposed to encode statistically independent features over a set of inputs. But it was in the 90´s when Bell and Sejnowski applied this theoretical concept to the blindly separation of the mixed sources (BSS) using a well known stochastic gradient learning rule [2] and originating a productive period of research in this area [3–6]. In this way ICA algorithms have been applied successfully to several fields such as biomedicine, speech, sonar and radar, signal processing, etc. and more recently also to time series forecasting [7], i.e. using stock data [8]. In the latter application the mixing process of multiple sensors is based on linear transformation making the following assumptions:

1. the original (unobservable) sources are statistically independent which are related to social-economic events.
2. the number of sensors (stock series) is equal to that of sources.
3. the Darmois-Skitovick conditions are satisfied [9].

C.G. Puntonet and A. Prieto (Eds.): ICA 2004, LNCS 3195, pp. 414–421, 2004.

On the other hand there is a wide class of interesting applications for which no reasonably fast algorithms have been developed, i.e. optimization problems that appear frequently in several applications such as VLSI design or the travelling salesman problem. In general, any abstract task to be accomplished can be viewed as a search through a space of potential solutions and whenever we work with large spaces, GAs are suitable artificial intelligence techniques for developing this optimization [10, 11]. GA are stochastic algorithms whose search methods model some natural phenomena according to genetic inheritance and Darwinian strife for survival. Such search requires balancing two goals: exploiting the best solutions and exploring the whole search space. In order to carry out them GA performs an efficient multi-directional search maintaining a population of potential solutions instead of methods such as simulated annealing or Hill Climbing.

In this work we apply GA to ICA in the search of the separation matrix, in order to improve the performance of endogenous learning machines in real time series forecasting speeding up convergence rates (scenarios with the BSS problem in higher dimension). We organize the essay as follows. In section 2 we give a brief overview of the basic GA theory and introduce a set of new genetic operators in sections 3 and 4. The new search algorithm will be compare to the well-known ICA algorithms and state state some conclusions in section 5.

2 Basis Genetic Algorithms in Higher Dimension

A GA can be modelled by means of a *time inhomogeneous Markov* chain [12] obtaining interesting properties related with weak and strong ergodicity, convergence and the distribution probability of the process [13]. In the latter reference, a canonical GA is constituted by operations of parameter encoding, population initialization, crossover , mutation, mate selection, population replacement, fitness scaling, etc. proving that with these simple operators a GA does not converge to a population containing only optimal members. However, there are GAs that converge to the optimum, *The Elitist GA* [14] and those which introduce *Reduction Operators*[15]. We have borrowed the notation mainly from [13] where the model for GAs is a inhomogeneous Markov chain model on probability distributions ($\mathbf{S}$) over the set of all possible populations of a fixed finite size. Let $\mathbf{C}$ the set of all possible creatures in a given world (number of vectors of genes equal to that of elements of the mixing matrix) and a function $f : \mathbf{C} \to R^+$ (see section 2.1). The task of GAs is to find an element $c \in \mathbf{C}$ for which $f(c)$ is maximal. We encode creatures into genes and chromosomes or individuals as strings of length ℓ of binary digits (size of Alphabet A is $a = 2$) using one-complement representation.

In the Initial Population Generation step (choosing randomly $p \in \wp_N$, where $\wp_N$ is the set of populations, i.e the set of N-tuples of creatures containing $a^{L \equiv N \cdot \ell}$ elements) we assume that creatures lie in a bounded region $[-1, 1]$. After the initial population p has been generated, the fitness of each chromosome $\mathbf{c}_i$ is determined using a contrast function (i.e based on cumulants or neg-entropy) which measures the pair-wise statistical independency between sources in the current individual (see section 2.1).

Table 1. Pseudo-code of GA.

```
Initialize Population
i=0
while not stop do
  do N/2 times
    Select two mates from p_i
    Generate two offspring using crossover operator
    Mutate the two children
    Include children in new generation p_new
  end do
  Build population p̂_i = p_i ∪ p_new
  Apply Reduction Operators (Elitist Strategies) to get p_{i+1}
  i=i+1
end
```

The next step in canonical GA is to define the Selection Operator. New generations for mating are selected depending on their fitness function values using *roulette wheel selection*. Let $p = (c_1, \ldots, c_N) \in \wp_N$, $n \in \mathcal{N}$ and f the fitness function acting in each component of p. Scaled fitness selection of p is a lottery for every position $1 \leq i \leq N$ in population p such that creature c_j is selected with probability proportional to its fitness value. Thus proportional fitness selection can be described by column stochastic matrices $\mathbf{F_n}$, $n \in \mathcal{N}$, with components:

$$\langle q, \mathbf{F_n} p \rangle = \prod_{i=1}^{N} \frac{n(q_i) f_n(p, q_i)}{\sum_{j=1}^{N} f_n(p, j)} \tag{1}$$

where $p, q \in \wp_N$ so $p_i, q_i \in \mathbf{C}$, $\langle \ldots \rangle$ denotes the standard inner product, and $n(q_i)$ the number of occurrences of q_i in p. Once the two individuals have been selected, an elementary crossover operator $\mathbf{C}(K, P_c)$ is applied (setting the crossover rate at a value, i.e. $P_c \to 0$, which implies children similar to parent individuals) that is given (assuming N even) by:

$$\mathbf{C}(K, P_c) = \prod_{i=1}^{N/2} ((1 - P_c)\mathcal{I} + P_c \mathbf{C}(2i - 1, 2i, k_i)) \tag{2}$$

where $\mathbf{C}(2i-1, 2i, k_i)$ denotes elementary crossover operation of c_i, c_j creatures at position $1 \leq k \leq \ell$ and $\mathcal{I}$ the identity matrix, to generate two offspring (see [13] for further properties of the crossover operator), $K = (k_1, \ldots, k_{N/2})$ a vector of cross over points and P_c the cross over probability.

2.1 Fitness Function Based on Cumulants

The independence condition for the independent components of the output vector $\mathbf{y}$ is given by the definition of independence random variables:

$$p(\mathbf{y}) = \prod_{i=1}^{n} p_{y_i}(y_i); \tag{3}$$

In order to measure the independence of the outputs we express equation 3 in terms of higher order statistics (cumulants) using the characteristic function (or moment generating function) $\phi(\mathbf{k})$, where $\mathbf{k}$ is a vector of variables in the Fourier transform domain, and considering its natural logarithm $\Phi = log(\phi(\mathbf{k}))$. Thus we get:

$$Cum(\overbrace{y_i, y_j, \ldots}^{stimes}) = \kappa_s^i \delta_{i,j,\ldots} \qquad \forall i, j, \ldots \in [1, \ldots, n] \tag{4}$$

where $Cum(\overbrace{\ldots}^{stimes})$ is the s-th order cross-cumulant and $\kappa_s = Cum(\overbrace{y_i}^{stimes})$ is the auto-cumulant of order s straightforward related to moments [16]. Hence vanishing cross-cumulants are a necessary condition for $y_1, \ldots, y_n$ to be independent[1]. Based on the briefly above discussion, we can define the fitness function for BSS as:

$$f(p_o) = \sum_{i,j,\ldots} ||Cum(\overbrace{y_i, y_j, \ldots}^{stimes})|| \qquad \forall i, j, \ldots \in [1, \ldots, n] \tag{5}$$

where p_o is the parameter vector (individual) containing the separation matrix and $||\ldots||$ denotes the absolute value.

3 Mutation Operator Based on Neighborhood Philosophy

The new Mutation Operator $\mathbf{M_{P_m}}$ is applied (with probability $\mathbf{P}_m$) independently at each bit in a population $p \in \wp_N$, to avoid premature convergence (see [10] for further discussion) and enforcing strong ergodicity. The multi-bit mutation operator with changing probability following a *exponential* law with respect to the position $1 \leq i \leq L$ in $p \in \wp_N$:

$$P_m(i) = \mu \cdot \exp\left(\frac{-mod\{\frac{i-1}{N}\}}{\emptyset}\right) \tag{6}$$

where $\emptyset$ is a normalization constant and μ the change probability at the beginning of each creature p_i in population p; can be described as a positive stochastic matrix in the form:

$$\langle q, \mathbf{M_{P_m}} p\rangle = \mu^{\Delta(p,q)} \exp\left(-\sum_{dif(i)}^{\Delta(p,q)} \frac{mod\{\frac{i-1}{N}\}}{\emptyset}\right) \cdot \prod_{equ(i)}^{L-\Delta(p,q)} [1 - P_m(i)] \tag{7}$$

where $\Delta(p,q)$ is the Hamming distance between p and $q \in \wp_N$, $dif(i)$ resp. $equ(i)$ is the set of indexes where p and q are different resp. equal. Following from equation 7 and checking how the matrices act on populations we can write:

$$\mathbf{M_{P_m}} = \prod_{\lambda=1}^{N} \left([1 - P_m(i)]\,\mathbf{1} + P_m(i)\hat{\mathbf{m}}^1(\lambda)\right) \tag{8}$$

[1] In practice we need independence between sources two against two.

where $\hat{\mathbf{m}}^1(\lambda) = \mathbf{1} \otimes \mathbf{1} \ldots \otimes \overbrace{\hat{m}^1}^{\lambda} \otimes \ldots \otimes \mathbf{1}$ is a linear operator on $V_\wp$, the free vector space over A^L and $\hat{m}^1$ is the linear 1-bit mutation operator on V_1, the free vector space over A. The latter operator is defined acting on Alphabet as:

$$\langle \hat{a}(\tau'), \hat{m}^1 \hat{a}(\tau) \rangle = (a-1)^{-1}, \quad 0 \leq \tau' \neq \tau \leq a-1 \tag{9}$$

i.e. probability of change a letter in the Alphabet once mutation occurs with probability equal to $L\mu$. The spectrum of $\mathbf{M}_{\mathbf{P_m}}$ can be evaluated according to the following expression:

$$sp(\mathbf{M}_{\mathbf{P_m}}) = \left\{ \left(1 - \frac{\mu(\lambda)}{a-1} \right)^{\lambda}; \quad \lambda \in [0, L] \right\} \tag{10}$$

where $\mu(\lambda) = \exp\left(\frac{-mod\{\frac{\lambda-1}{N}\}}{\emptyset} \right)$.

The operator presented in equation8 has similar properties to the Constant Multiple-bit mutation operator $\mathbf{M}_\mu$ presented in [13]. $\mathbf{M}_\mu$ is a contracting map in the sense presented in [13]. It is easy to prove that $\mathbf{M}_{\mathbf{P_m}}$ is a contracting map too, using the Corollary B.2 in [13] and the eigenvalues of this operator(equation 10). We can also compare the coefficients of ergodicity:

$$\tau_r(\mathbf{M}_{\mathbf{P_m}}) < \tau_r(\mathbf{M}_\mu) \tag{11}$$

where $\tau_r(\mathbf{X}) = max\{\|\mathbf{X}v\|_r : \quad v \in \mathcal{R}^n, \quad v \perp e \quad and \quad \|v\|_r = 1\}$.

Mutation is more likely at the beginning of the string of binary digits ("small neighborhood philosophy"). In order to improve the speed convergence of the algorithm we have included mechanisms such as elitist strategy (reduction operator [17] consisting of sampling a Boltzmann probability distribution in the extended population) in which the best individual in the current generation always survives into the next (a further discussion about reduction operator, $\mathbf{P}_R$, can be found in [18]).

4 Guided Genetic Algorithm

In order to include statistical information into the algorithm (it would be a nonsense to ignore it!) we define the hybrid statistical genetic operator based on reduction operators as follows (in standard notation acting on populations):

$$\langle q, \mathbf{M}^{\mathbf{n}}_{\mathbf{G}} p \rangle = \frac{1}{\aleph(T_n)} \exp\left(-\frac{||q - \mathbf{S}^{\mathbf{n}} \cdot p||^2}{T_n} \right); \qquad p, q \in \wp_N \tag{12}$$

where $\aleph(T_n)$ is the normalization constant depending on temperature T_n, n is the iteration and $\mathbf{S}^{\mathbf{n}}$ is the step matrix which contains statistical properties, i.e based on cumulants it can be expressed using quasi-Newton algorithms as [5]:

$$\mathbf{S}^{\mathbf{n}} = (\mathbf{I} - \mu^n(\mathbf{C}^{1,\beta}_{y,y}\mathbf{S}^{\beta}_{y} - \mathbf{I})); \qquad p_i \in \mathbf{C} \tag{13}$$

where $\mathbf{C}_{y,y}^{1,\beta}$ is the cross-cumulant matrix whose elements are $[\mathbf{C}_{y,y}^{\alpha,\beta}]_{ij} = Cum(\underbrace{y_i,\ldots,y_i}_{\alpha},\underbrace{y_j,\ldots,y_j}_{\beta})$ and $\mathbf{S}_y^\beta$ is the sign matrix of the output cumulants.

Such search requires balancing two goals: exploiting the blindly search like a canonical GA and using statistical properties like a standard ICA algorithm. Finally the guided GA (GGA) is modelled, at each step, as the stochastic matrix product acting on probability distributions over populations:

$$\mathbf{G^n} = \mathbf{P}_R^n \cdot \mathbf{F_n} \cdot \mathbf{C}_{\mathbf{P_c^n}}^{\mathbf{k}} \cdot \mathbf{M}_{(\mathbf{P_m},\mathbf{G})^\mathbf{n}} \tag{14}$$

The GA used applies local search (using the selected mutation and crossover operators) around the values (or individuals) found to be optimal (elite) the last time. The computational time depends on the encoding length, number of individuals and genes. Because of the probabilistic nature of the GA-based method, the proposed method almost converges to a global optimal solution on average. In our simulation, however, nonconvergent case was not found. Table 1 shows the GA-pseudocode.

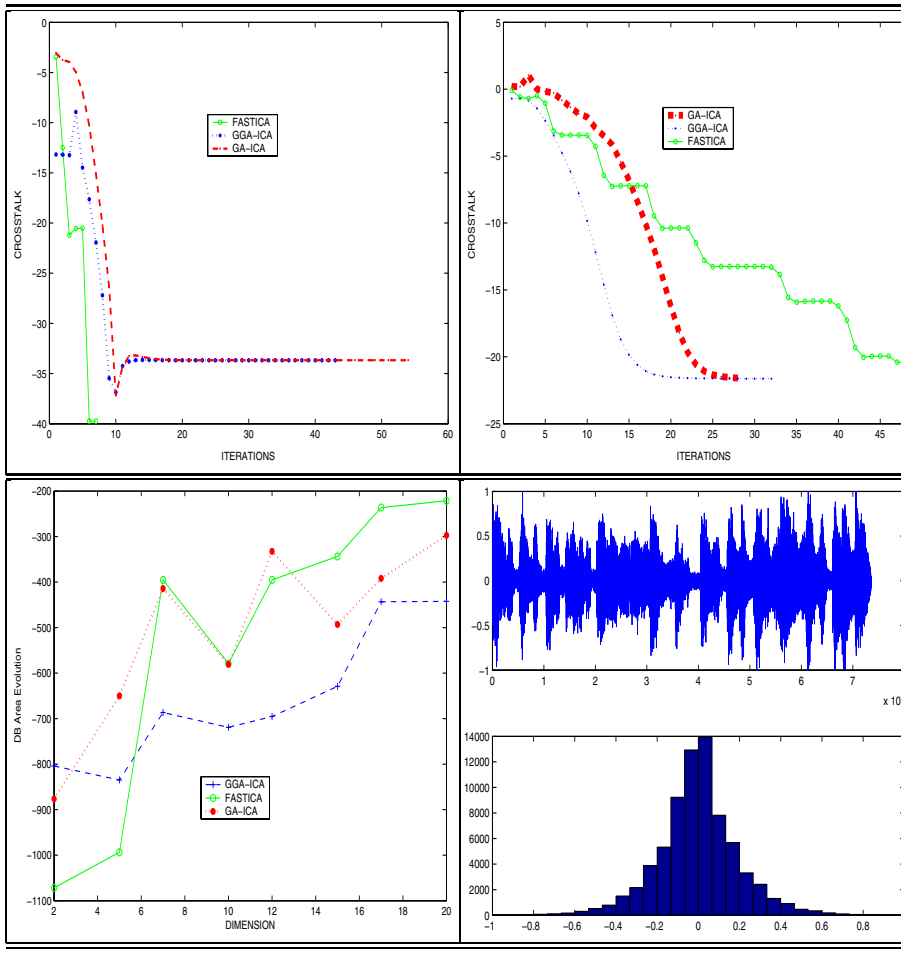
5 Simulations and Conclusions

To check the performance of the proposed hybrid algorithm, 50 computer simulations were conducted to test the GGA vs. the GA method [7] and the most relevant ICA algorithm to date, FastICA [5]. In this paper we neglect the evaluation of the computational complexity of the current methods, described in detail in several references such as [7] or [19]. The main reason lies in the fact that we are using a 8 nodes Cluster Pentium II 332MHz 512Kb Cache, thus the computational requirements of the algorithms (fitness functions, encoding, etc.) are generally negligible compared with the cluster capacity. Logically GA-based BSS approaches suffer from a higher computational complexity.

Consider the mixing cases from 2 to 20 independent random super-gaussian input signals. We focuss our attention on the evolution of the crosstalk vs. the number of iterations using a mixing matrix randomly chosen in the interval $[-1,+1]$. The number of individuals chosen in the GA methods were $N_p = 30$ in the 50 (randomly mixing matrices) simulations for a number of input sources from 2 (standard BSS problem) to 20 (BSS in biomedicine or finances). The standard deviation of the parameters of the separation over the 50 runs never exceeded 1% of their mean values while using the FASTICA method we found large deviations from different mixing matrices due to its limited capacity of local search as dimension increases. The results for the crosstalk are displayed in Table 2. It can be seen from the simulation results that the FASTICA convergence rate decreases as dimension increases whereas GA approaches work efficiently.

A GGA-based BSS method has been developed to solve BSS problem from the linear mixtures of independent sources. The proposed method obtain a good performance overcoming the local minima problem over multidimensional domains. Extensive simulation results prove the ability of the proposed method.

Table 2. Figures: 1) Mean Crosstalk (50 runs) vs. iterations to reach the convergence for num. sources equal to 2 2) Mean Crosstalk (50 runs) vs. iterations to reach the convergence for num. sources equal to 20 3) Evolution of the crosstalk area vs. dimension. 4) Example of independent source used in the simulations.



This is particular useful in some medical applications where input space dimension increases and in real time applications where reaching fast convergence rates is the major objective.

References

1. Barlow, H.B, Possible principles underlying transformation of Sensory messages. Sensory Communication, W.A. Rosenblith, MIT Press, New York, U.S.A. (1961).
2. Bell, A.J., Sejnowski, T.J. An Information-Maximization Approach to Blind Separation and Blind Deconvolution. Neural Computation, vol 7, 1129-1159 (1995).

3. Cardoso, J.F., Infomax and maximun likelihood for source separation. IEEE Letters on signal processing, 4, 112-114 (1997).
4. Cichoki, A., Unbehauen, R., Robust neural networks with on-line learning for blind identification and blind separation of sources. IEEE Transactions on Circuits and Systems, 43 (11), 894-906 (1996).
5. Hyvärinen, A., Oja, E., A fast fixed point algorithm for independent component analysis Neural Computation, 9: 1483-1492
6. Puntonet, C.G., Prieto, A. Neural net approach for blind separation of sources based on geometric properties. Neurocomputing 18, 141-164 (1998)
7. Górriz, J.M., Algorítmos Híbridos para la Modelización de Series Temporales con Técnicas AR-ICA. PhD Thesis, University of Cádiz (2003)
8. Back, A.D., Weigend, A.S., A first Application of Independent Component Analysis to Extracting Structure from Stock Returns. International Journal of Neural Systems, vol 8,(5), (1997)
9. Cao, X.R., Liu W.R., General Approach to Blind Source Separation. IEEE Transactions on signal Processing, vol 44, num 3, 562-571 (1996)
10. Michalewicz, Z., *Genetic Algorithms + Data structures = Evolution Programs*, springer Verlag, Berlin 1992.
11. Rojas F., Álvarez M.R., Puntonet C.G., Martin-Clemente R., Applying Neural Networks and Genetic Algorithms to the Separation of Sources Iberama 2002 LNAInteligence 2527,420-429, Sevilla (2002)
12. Haggstrom, O., *Finite Markov Chains and Algorithmic Applications*, Cambridge University,1998.
13. Schmitt, L.M., Nehaniv, C.L., Fujii, R.H., *Linear Analysis of Genetic Algorithms*, Theoretical Computer Science, volume 200, pages 101-134, 1998.
14. Suzuki, J., *A markov Chain Analysis on Simple Genetic Algorithms*, IEEE Transaction on Systems, Man, and Cybernetics, vol 25, 4, 655-659,(1995).
15. Eiben, A.E., Aarts, E.H.L., Van Hee, K.M., Global Convergence of Genetic Algorithms: a Markov Chain Analysis, Parallel Problem Solving from Nature, Lecture Notes in Computer Science, vol 496, (4-12),(1991).
16. Chryssostomos, C., Petropulu, A.P., Higher Order Spectra Analysis: A Non-linear Signal Processing Framework Prentice Hall, London (1993)
17. Lozano, J.A., Larrañaga, P., Graña, M., Albizuri, F.X., *Genetic Algorithms: Bridging the Convergence Gap*, Theoretical Computer Science, vol 229, 11-22, (1999).
18. Rudolph, G., *Convergence Analysis of Canonical Genetic Algorithms*, IEEE Transactions on Neural Networks, vol 5, num 1,(1994) 96-101.
19. Tan, Y., Wang, J., Nonlinear Blind Source Separation Using Higher order Statistics and a Genetic Algorithm. IEEE Transactions on Evolutionary Computation, vol. 5, num 6 (2001)